German Research School
for Simulation Sciences

Stochastic Mode Sampling (SMS) - An Efficient Approach to the Analytic Continuation Problem

Master's Thesis

Khaldoon Ghanem

October 2013

Supervisor
Prof. Dr. Erik Koch

Examiner
Prof. Dr. Erik Koch

Co-Examiner
Prof. Dr. Eva Pavarini

إلى أمي وأبي الغاليين
إلى سورية الجريحة

To my dear parents
To bleeding Syria

Acknowledgements

I would like to thank my supervisor Prof. Erik Koch for providing guidance and knowledge throughout the entire thesis. His door was always open for my frequent questions, and our long discussions taught me how to think as a scientist. I am deeply grateful for that!

I would like also to thank the German Academic Exchange Service (DAAD) for their financial support during the whole Master's program. Without their scholarship, I would not be able to pursue my graduate studies in Germany.

Abstract

Analytic continuation is a recurring problem in different contexts of condensed matter physics. Typically we need to find a non-negative function, like spectral function or optical conductivity, using data from Quantum Monte Carlo (QMC) simulations. The relation between the data and the desired function can be formulated as a Fredholm integral equation of first kind which is an ill-posed problem with no unique solution in the presence of noise. What is special about these particular Fredholm integral equations is the non-negativity of the solution. Utilizing this property does not only make sure we get a physically-acceptable solution, but it also provides additional information that helps improving its quality.

One class of methods that solve the problem using only the non-negativity of the solution as a priori knowledge, is the *Stochastic Sampling*. These methods use Bayesian inference to derive a probability distribution of the solution and use the mean as an estimator. To get the mean, they usually sample the solution space directly, but unfortunately this leads to large correlation time due to the high correlations between the different components of the solution.

In this thesis, we propose a new stochastic sampling method, *Stochastic Mode Sampling* (SMS), where instead of sampling the solution's components directly, we sample the right singular vectors (modes) of the kernel of the integral equation using Gibbs sampling. In this basis, the sampled quantities are statistically uncorrelated, but they are coupled through the non-negativity constraint. The efficiency of our method depends on this coupling, so we also show how to modify the kernel and choose the grid such that the coupling is minimized. Using the proper modification and grid, the SMS method has much less correlation times than earlier stochastic sampling methods. Besides, since the modes are ordered naturally according to their relevance to the data (using singular values), the SMS method provides a convenient way of trading-off between quality and speed by simply limiting the modes included in the sampling.

Contents

CHAPTER 1

Introduction

Analytic continuation, as used in mathematics, refers to extending the domain of a complex function. It provides a way of obtaining the values of a complex function in some region knowing its values in another. This is a recurring problem in different contexts of condensed matter physics because quantum Monte Carlo (QMC) simulations often produce results on the imaginary axis. Those results then need to be analytically continued to the real axis in order to compute the dynamical properties of the physical system of interest.

One example of analytic continuation is obtaining the spectral function $A(\omega)$ at real frequencies from Green function $\mathcal{G}$ values either at Matsubara frequencies $i\omega_n$ or imaginary time τ. The Green and spectral functions are related by the following equivalent[1] relations

$$\mathcal{G}(i\omega_n) = \int d\omega \, \frac{1}{i\omega_n - \omega} \quad A(\omega), \quad n = 0, 1, 2, ... \tag{1.1}$$

$$\mathcal{G}(\tau) = \int d\omega \, \frac{-e^{-\tau\omega}}{1 \pm e^{-\beta\omega}} \, A(\omega), \quad \tau \in [0, \beta] \tag{1.2}$$

where the upper (lower) sign is for fermionic (bosonic) case, $\omega_n = (2n+1)\pi/\beta \quad [2n\pi/\beta]$ for fermionic [bosonic] case, and $\beta = 1/T$ is the inverse temperature. Details on the origin of the previous relations can be found in App. A.

Another example is obtaining the optical conductivity function $\sigma(\omega)$ from the current-current correlation function Π which can also be provided at either Matsubara frequencies or imaginary time. The relations between the two functions for the bosonic case are

$$\Pi(i\omega_n) = \frac{1}{\pi} \int d\omega \, \frac{\omega^2}{\omega_n^2 + \omega^2} \quad \sigma(\omega), \quad n = 0, 1, 2, ... \tag{1.3}$$

$$\Pi(\tau) = \frac{1}{\pi} \int d\omega \, \frac{\omega e^{-\tau\omega}}{1 - e^{-\beta\omega}} \, \sigma(\omega), \quad \tau \in [0, \beta] \tag{1.4}$$

[1] They are Fourier transforms of each other.

A key feature of these analytic continuation relations is that they can always be rewritten such that the unknown function is non-negative. For example, the fermionc spectral function $A(\omega)$ is itself non-negative so nothing needs to be done. The bosonic spectral function, however, does not satisfy this property but rather $A(\omega)/\omega \geq 0$. Then we can, for example, rewrite Eq. (1.1) for the bosonic case as

$$\mathcal{G}(i\omega_n) = \int d\omega \, \frac{\omega}{i\omega_n - \omega} \, \frac{A(\omega)}{\omega}. \tag{1.5}$$

Non-negativity will play an important role in solving the analytic continuation problem later. A simplistic argument for the usefulness of the non-negativity can be formulated as following. Suppose the unknown function is represented using only two values, then the function lives in a plane. Restricting our attention to the non-negative functions only, reduces the space to the positive quadrant. Generally, if the function is represented using n values, then the space size is reduced by a factor of $1/2^n$; a tremendous reduction!

Generally, the analytic continuation problem can be formulated as a Fredholm integral equation of first kind

$$g(y) = \int K(y,x) f(x) dx \tag{1.6}$$

where $K(y,x)$ is called the integral kernel, $g(y)$ is the data and $f(x)$ the model. The goal is to find the model given the data, the kernel and any prior information about the model itself (e.g. non-negativity).

Fredholm integral equations are well-known beyond the analytic continuation problem and have applications in many different fields. The techniques presented here can thus also be used in solving other problems; the only difference is in the kernel form and the prior knowledge about the solution when such knowledge is available.

The difficulty in solving Fredholm integral equations is that they are inherently ill-posed. When we compute the data, sharp features in the model get smoothed and errors get damped due to the integration. The inverse process, on the other hand, is problematic; small errors in the data may lead (depending on the used method) to very large errors in the reconstructed model. We need to mention that the ill-posed nature is not related to the kernel form but rather to the fact that the model belongs to an infinite dimensional space (see Ref. [1], Theorem 15.4).

In Ch. 2, we present different approaches for solving the analytic continuation problem including our new method, the stochastic modes sampling (SMS). In Ch. 3, we study in detail the application of SMS method to a specific case. App. A provides some background information on Green functions and their analytic continuation, Eqs. (1.1) and (1.2).

CHAPTER 2

Solving the Analytic Continuation Problem

As described in the introduction, the analytic continuation problem is mathematically solving a Fredholm integral equation of first kind

$$\int K(y,x)f(x)dx = g(y) \tag{2.1}$$

where the kernel $K(y,x)$ is known analytically, a finite number of data values $g(y)$ are available, and we need to find the model $f(x)$ which is known to be non-negative. Usually the data is only known with some uncertainty because of the noise on it, and this is what makes the problem difficult.

In this chapter, we present different numerical methods. First, we discretize the integral in the previous equation to bring it into a form suitable for numerical calculations. Then we describe early attempts in solving the problem. Finally, we present the stochastic sampling method and our new approach to it, Stochastic Mode Sampling.

2.1. Discretization

The first step in solving Eq. (2.1) is to discretize it and obtain an approximate algebraic equation. The discretization of the y coordinate is already determined by the available data values. We assume there are m such values $g(y_j)$ and organize them in a column vector $\mathbf{G} \in \mathbb{R}^m$. For discretizing the left-hand side, we have two options:

Numerical Quadrature We introduce a grid[1] in the variable x and evaluate the integral using some numerical quadrature

$$\int K(y_j,x)f(x)dx \approx \sum_{i=1}^{n} w_i K(y_j,x_i)f(x_i)\ . \tag{2.2}$$

We then build a column vector $\mathbf{F} \in \mathbb{R}^n$ whose elements are $\sqrt{w_i}f(x_i)$ and a matrix $\mathbf{K} \in \mathbb{R}^{m\times n}$ whose elements are $\sqrt{w_i}K(y_j,x_i)$. The weights could be removed from F and included entirely in the matrix. However, splitting them in the earlier way has the advantage of using the euclidean norm of $\mathbf{F}$ as an approximation of the L^2-norm of $f(x)$

$$\mathbf{F}^T\mathbf{F} \approx \int |f(x)|^2 dx\ . \tag{2.3}$$

Galerkin Method If we expect that the model can be well-approximated using some finite function basis, we can expand it in that basis

$$f(x) \approx \sum_{i=1}^{n} f_i \phi_i(x) \tag{2.4}$$

[1] Suitable gird cutoff and spacing are not known beforehand. Typically one would start with a small coarse grid and then refine it until no important changes in the reconstructed model are observed.

where f_i are the expansion coefficients. Then we compute the integrals of the kernel with each of the basis functions either analytically, if possible, or numerically up to the machine accuracy

$$\int K(y_j, x)\phi_i(x) = \mathbf{K}_{i,j} \ . \tag{2.5}$$

Finally we organize the expansion coefficients in a column vector $\mathbf{F} \in \mathbb{R}^n$ and basis function integral values in a matrix $\mathbf{K} \in \mathbb{R}^{m \times n}$.

Both of the above mentioned methods, give us a system of linear equations:

$$\begin{aligned} \mathbf{K}\ \mathbf{F} = \mathbf{G}, \quad \text{with} \quad & \mathbf{F} \in \mathbb{R}^n \\ & \mathbf{G} \in \mathbb{R}^m \\ & \mathbf{K} \in \mathbb{R}^{m \times n} \ , \end{aligned} \tag{2.6}$$

and the problem is finding the unknown vector $\mathbf{F}$.

Tip It is worth noting that when the integral extends from $-\infty$ to $+\infty$ (which is usually the case for analytic continuation) and the integrand is analytic in an open strip around the real axis, then it is recommended to use the trapezoidal rule for discretizing the integral. Besides the simplicity of this rule, it converges exponentially with the grid spacing if the above mentioned conditions are satisfied (see Refs. [2, 3]). For an example, see the case study in Ch. 3.

Complex Case In case of a complex kernel and complex data, like Eq. (1.1), we can still represent the problem in real space. We split the real and imaginary part of both the kernel and the data

$$\int [K_1(y, x) + i\ K_2(y, x)] f(x) dx = g_1(y) + i\ g_2(y) \ . \tag{2.7}$$

Since we assume the model is non-negative, it is implicitly real and thus the real part of the data is generated by the real part of the kernel only and the imaginary part of the data is generated by the imaginary part of the data only

$$\int K_1(y, x) f(x) dx = g_1(y) \tag{2.8}$$

$$\int K_2(y, x) f(x) dx = g_2(y) \ . \tag{2.9}$$

This way the original complex equation is equivalent to two real decoupled ones. These real equations can then be discretized as discussed earlier resulting in a real system of linear equations

$$\begin{aligned} \mathbf{K}\ \mathbf{F} = \begin{bmatrix} \mathbf{K_1} \\ \mathbf{K_2} \end{bmatrix} \ \mathbf{F} = \begin{bmatrix} \mathbf{G_1} \\ \mathbf{G_2} \end{bmatrix} = \mathbf{G}, \quad \text{with} \quad & \mathbf{F} \in \mathbb{R}^n \\ & \mathbf{G} \in \mathbb{R}^{2m} \\ & \mathbf{K} \in \mathbb{R}^{2m \times n} \ , \end{aligned} \tag{2.10}$$

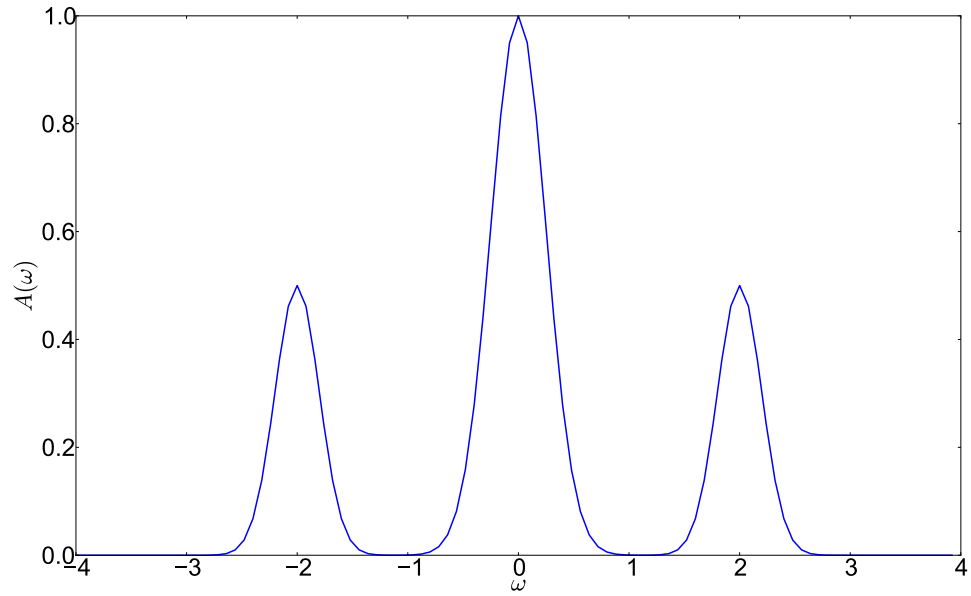

Fig. 2.1.: The spectral function (model) of test case.

Test Case The following setup will be used throughout this chapter as a test case illustrating different concepts. We address the analytic continuation of fermionic imaginary-time Green function described by Eq. (1.2). The spectral function $A(\omega)$ is the one shown in Fig. 2.1 and the inverse temperature is $\beta = 50$. The integral is discretized using the trapezoidal rule on a uniform grid form -4 to +4 with spacing of 0.08. Green function values are generate at 600 equally-spaced τ points in the interval $[0, \beta]$. To simulate the effect of computational errors existing in QMC data, we put noise on the data using the relation $\tilde{\mathcal{G}}(\tau_j) = \mathcal{G}(\tau_j) * (1 + r_j)$ where r_j are normal random variables with zero mean and variance $\sigma = 10^{-4}$.

2.2. Least Squares Solution

Typically the matrix $\mathbf{K}$ is rank deficient, so the range of $\mathbf{K}$ does not cover the whole $\mathbb{R}^m$. Since actual data obtained by simulation or measurement contains noise, it will lie outside the range of $\mathbf{K}$, and in general there is no model $\mathbf{F}$ satisfying Eq. (2.6) exactly. The standard approach then is to find the model which produces the data closest to the actual one in a least squares sense. Such a solution is called *least squares solution* and it is defined as the model minimizing the residual (the euclidean distance between the actual data and the data produced by the model):

$$\mathbf{F}_{\mathrm{LS}} = \arg\min_{\mathbf{F}\in\mathbb{R}^n} \| \mathbf{K}\,\mathbf{F} - \mathbf{G} \|_2 \ . \tag{2.11}$$

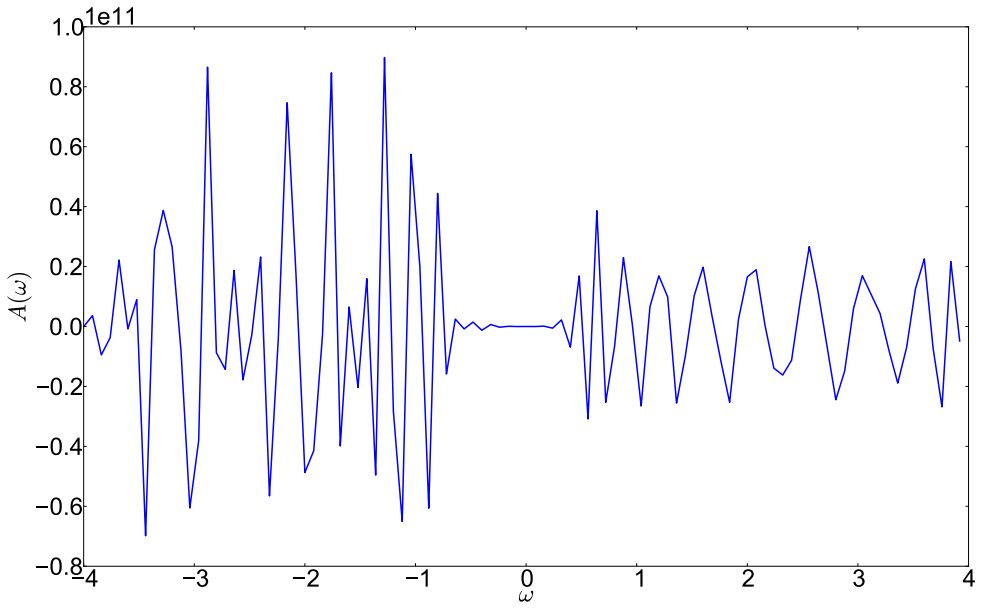

Fig. 2.2.: The spectral function reconstructed using the least squares method. Sawtooth noise is dominating and no useful information can be extracted.

$\mathbf{F}_{\mathrm{LS}}$ is found by setting the derivative of the residual to zero

$$\frac{d}{d\mathbf{F}} \parallel \mathbf{K}\ \mathbf{F} - \mathbf{G} \parallel_2 = 0 \Leftrightarrow \frac{d}{d\mathbf{F}} \parallel \mathbf{K}\ \mathbf{F} - \mathbf{G} \parallel_2^2 = 0 \Leftrightarrow \frac{d}{d\mathbf{F}^T}(\mathbf{F}^T\ \mathbf{K}^T - \mathbf{G}^T)(\mathbf{K}\ \mathbf{F} - \mathbf{G}) = 0$$
$$\Leftrightarrow \mathbf{K}^T\mathbf{K}\ \mathbf{F} = \mathbf{K}^T\mathbf{G}\ . \tag{2.12}$$

The last linear system is called the system of normal equations. When $\mathbf{K}$ has a full column rank, the normal equations have a unique solution. But when $\mathbf{K}$ is column rank deficient, which is typically the case, there are an infinite number of solutions with the same residual because adding a vector from the null space of $\mathbf{K}$ to a solution does not change its residual. So we add and the condition that the norm of $\mathbf{F}$ is minimal to define the solution uniquely. Chapter 5 of Ref. [4] discusses the least squares problem and several numerically stable algorithms for solving it.

Using the least squares solution for solving the system resulting from a Fredholm integral equation gives typically an extremely bad solution with extremely large sawtooth noise. The reason is that the matrix $\mathbf{K}$ has a very large condition number (a concept which will be discussed in next section).

Fig. 2.2 shows the least squares solution computed using the noisy data of the test case. The solution is totally dominated by sawtooth noise; the noise on the noise is of the order of 10^{11} even though the data noise is only of the order of 10^{-4}!

2.3. Singular Value Decomposition (SVD)

Every matrix $\mathbf{K} \in \mathbb{R}^{m \times n}$ can be decomposed as

$$\mathbf{K} = \mathbf{U}\ \mathbf{S}\ \mathbf{V}^T \tag{2.13}$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthogonal matrices [2], and $\mathbf{S} \coloneqq \mathrm{diag}(s_1, ..., s_p) \in \mathbb{R}^{m \times n},\ p = \min\{m, n\}$ with $s_1 \geq ... \geq s_p \geq 0$ is a diagonal matrix.

When $m < n$, S looks like this:

$$\begin{pmatrix} s_1 & & & & \\ & s_2 & & \emptyset & \\ & & \ddots & & \\ & \emptyset & & s_m & 0 \quad \cdots 0 \end{pmatrix}, \tag{2.14}$$

and when $m > n$, it looks like this:

$$\begin{pmatrix} s_1 & & & \\ & s_2 & \emptyset & \\ & & \ddots & \\ 0 & \cdots & 0 & s_n \\ & \emptyset & & \end{pmatrix}. \tag{2.15}$$

The diagonal elements of $\mathbf{S}$ are called the *singular values* of $\mathbf{K}$. Let r be the number of non-zero singular values, then the rank of $\mathbf{K}$ equals r and the *condition number* of $\mathbf{K}$ equals the ratio of the largest singular value to the smallest non-zero one, i.e.

$$\kappa(\mathbf{K}) = \frac{s_1}{s_r}\ . \tag{2.16}$$

Fig. 2.3 shows the singular values for our test case matrix. Note that they are exponentially decaying and drop below numerical accuracy after about 60 values. So r, the rank of this matrix, is about 60. Condition number is around 10^{15} which is the inverse of machine epsilon; this is typical for all large enough matrices resulting from discretizing Fredholm integral equations of first kind.

The columns of $\mathbf{V}$ (denoted by $\mathbf{v_i}$) form an orthonormal basis of $\mathbb{R}^n$; we call them the *right singular vectors* or *modes* for short. The modes corresponding to zero singular values form an orthonormal basis of the null space of $\mathbf{K}$; we call them the *free modes*. The columns of $\mathbf{U}$ (denoted by $\mathbf{u_i}$) form an orthonormal basis of $\mathbb{R}^m$; we call them the *left singular vectors*. These two sets of vectors are related by the following relation

$$\mathbf{K}\ \mathbf{v_i} = s_i\ \mathbf{u_i}, \quad i \in [1, p] \quad \text{where } p = \min\{m, n\}\ , \tag{2.17}$$

which we will use next to provide insight into the relation between the data and model.

[2] A matrix $\mathbf{Q}$ is orthogonal if and only if $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$

Let us expand the model in the orthonormal basis of the modes

$$\mathbf{F} = \sum_{i=1}^{n} (\mathbf{v_i}^T \mathbf{F})\ \mathbf{v_i}\ , \tag{2.18}$$

then, using Eq. (2.17), the corresponding data can be expressed as

$$\mathbf{G} = \mathbf{K}\ \mathbf{F} = \sum_{i=1}^{r} s_i\ (\mathbf{v_i}^T \mathbf{F})\ \mathbf{u_i}\ . \tag{2.19}$$

The projection coefficients of the data on $\mathbf{U}$ are the same as the projection coefficients of the model on $\mathbf{V}$ weighted by the singular values. Since the singular values are decaying (see Fig. 2.3), the coefficients $\mathbf{v_i^T F}$ of later modes and their associated noise are suppressed in comparison to the leading modes. The extreme case is a free mode (a mode corresponding to a zero singular value) which has no effect on the data whatsoever. So if we add any linear combination of free modes to a given model, it will produce the same data. This explain why the least squares problem does not have a unique solution when $r < n$.

Now let us look at the inverse problem. Given the data, we want to determine the corresponding model. When we expand the data in the orthonormal basis of left singular vectors

$$\mathbf{G} = \sum_{j=1}^{m} (\mathbf{u_j}^T \mathbf{G})\ \mathbf{u_j}\ , \tag{2.20}$$

the corresponding model is obtained utilizing Eq. (2.17)

$$\mathbf{F}_{\text{svd}} = \sum_{j=1}^{r} s_j^{-1}\ (\mathbf{u_j}^T \mathbf{G})\ \mathbf{v_j}\ . \tag{2.21}$$

Contrast to Eq. (2.19), contributions from the later coefficients $\mathbf{u_i^T G}$ and their associated noise get amplified in comparison to the leading coefficients. This discrepancy between the leading and later singular values increases as the condition number of the matrix $\mathbf{K}$ gets larger. The sum above goes only to r because the range of $\mathbf{K}$ is spanned only by the first r vectors $\mathbf{u_j}$. Exact data lies already within the range of the matrix so coefficients $\mathbf{u_j}^T \mathbf{G}$ when $j > r$ must be zero. For actual data, however, those coefficients are not zero in general. They are merely noise and cannot be produced by any model, so they must be truncated.

The previous operation can be expressed concisely as

$$\mathbf{F}_{\text{svd}} = \mathbf{K}^{+} \mathbf{G}\ . \tag{2.22}$$

where $\mathbf{K}^{+}$ is the pseudo inverse of $\mathbf{K}$, and it is defined as

$$\mathbf{K}^{+} \coloneqq \mathbf{V}\ \mathbf{S}^{+}\ \mathbf{U}^T \text{ with } \mathbf{S}^{+} \coloneqq \text{diag}(s_1^{-1}, ..., s_r^{-1}, 0, ..., 0) \in \mathbb{R}^{n \times n}\ . \tag{2.23}$$

Substituting Eq. (2.22) in Eq. (2.12), we see that $\mathbf{F}_{\text{svd}}$ satisfies the normal equations and thus it is actually a least squares solution. Besides, it is not just any least squares solution, it is the one with the minimal norm because all the free modes are set to zero.

In Fig. 2.4 we show the expansion coefficients for both the original model (Fig. 2.1) and the least squares solution (Fig 2.2). Notice how the least squares coefficients are similar for about the first twenty values but explode for later ones.

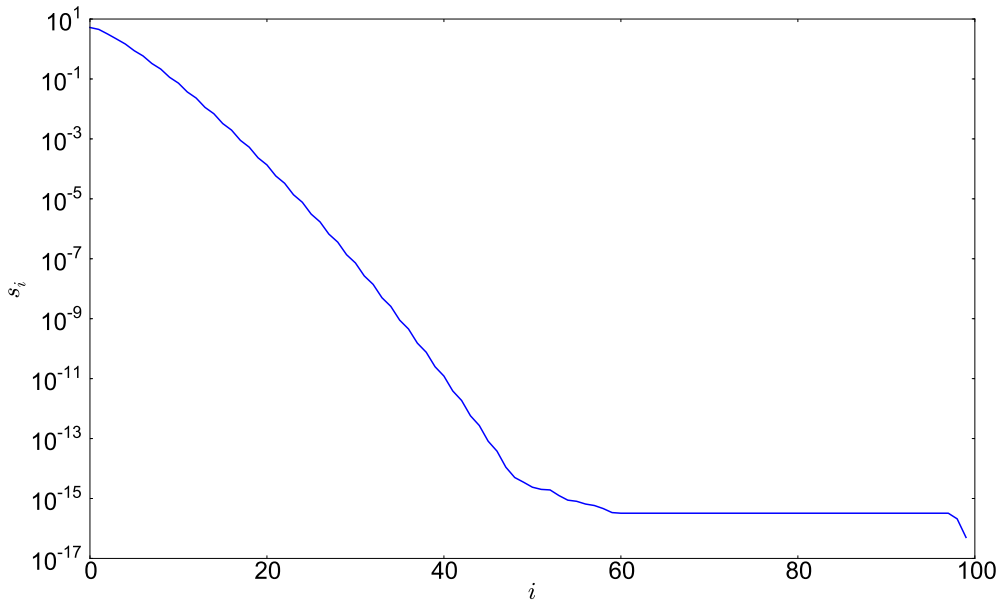

Fig. 2.3.: The singular values of the test case matrix on a semi-log plot. The singular values are computed using double precision float variables which have machine epsilon of 10^{-15}.

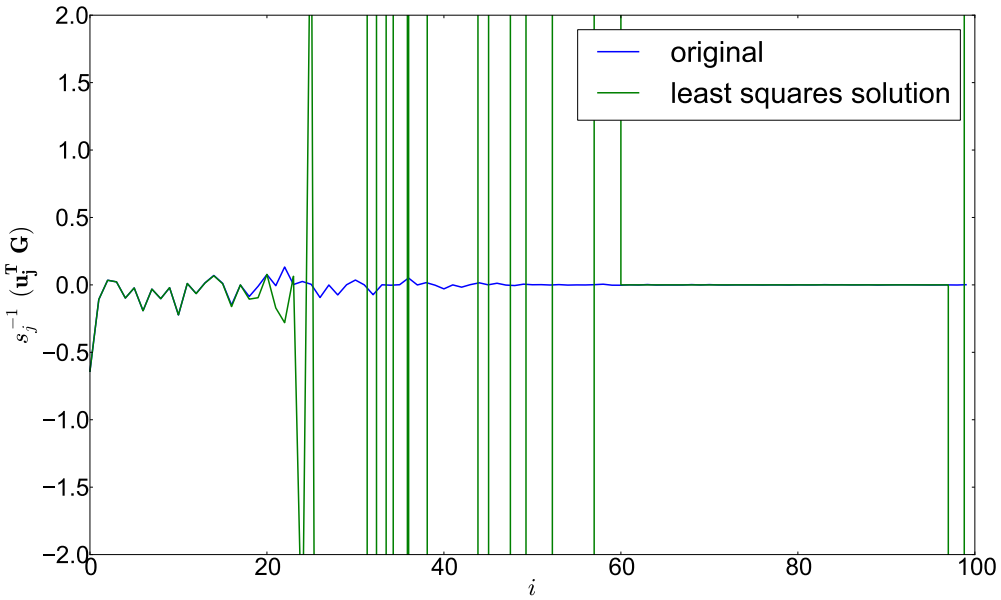

Fig. 2.4.: The expansion coefficients for both the original model (Fig. 2.1) and the least squares solution (Fig 2.2).

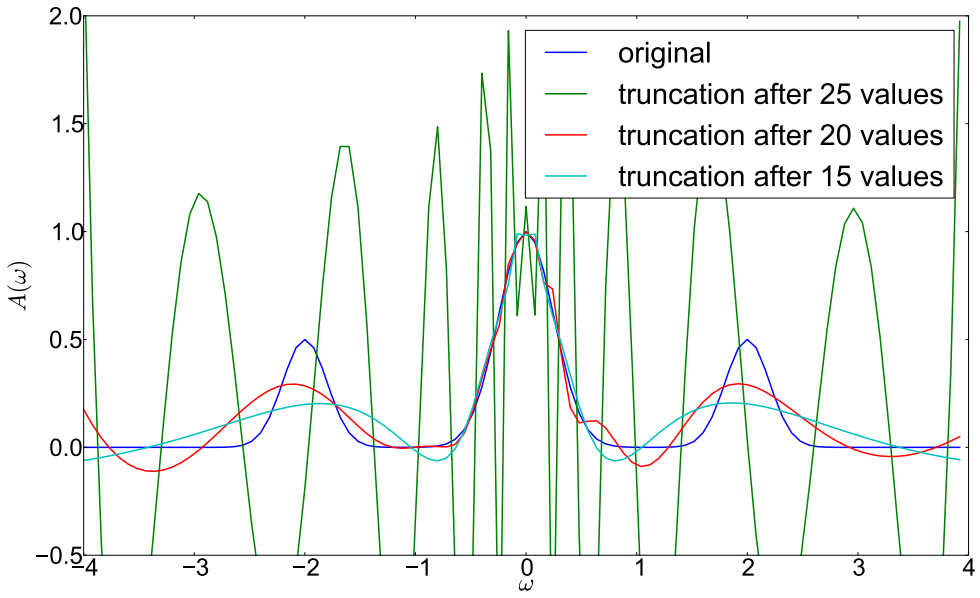

Fig. 2.5.: Truncated SVD solutions for different truncation limits. Compare it to Fig. 2.2.

2.4. Truncated SVD

As we have seen in the previous section, the sawtooth noise in the least squares solution comes from the late modes so one way to regularize the solution is to filter out those components

$$\mathbf{F}_{\text{TRUNC}} = \sum_{j=1}^{r} h_\alpha(s_j) s_j^{-1} \left(\mathbf{u_j}^T \mathbf{G}\right) \mathbf{v_j} \tag{2.24}$$

$$\text{with} \quad h_\alpha(s) = \begin{cases} 0 & \text{for } s \leq \alpha \\ 1 & \text{otherwise} \end{cases} \tag{2.25}$$

This truncation removes the noise associated with the truncated part but it also loses the information associated with it. The balance between these two effects can be tuned by the truncation parameter α. The larger α, the more values we truncate, the less noise we have and the more information we lose and vice versa. In Fig. 2.5, we show the solutions resulting from different truncations applied to out test case example. Notice that the more we truncate, the less the noise effect gets, and the smoother the solution becomes.

2.5. Tikhonov Regularization

Instead of using a filter function with a sharp cutoff, as in truncated SVD, one can use a smoothed version of it.

In Tikhonov regularization [5, 6], the terms are weighted as following

$$\mathbf{F}_{\mathrm{TIKH}} = \sum_{j=1}^{r} h_\alpha(s_j) s_j^{-1} \ (\mathbf{u_j}^T \mathbf{G}) \ \mathbf{v_j} \tag{2.26}$$

$$\text{with} \quad h_\alpha(s) = \frac{s^2}{s^2 + \alpha^2} \ . \tag{2.27}$$

Using this filter function, terms corresponding to very small singular values (and thus very large s_j^{-1}) are damped significantly ($\lim_{s\to 0} h_\alpha(s) = 0$), while the ones corresponding to large singular values are hardly modified ($\lim_{s\to\infty} h_\alpha(s) = 1$).

The Tikhonov solution, Eq. (2.26), can be found by solving the system

$$(\mathbf{K}^T\mathbf{K} + \alpha^2\mathbf{I}) \ \mathbf{F} = \mathbf{K}^T\mathbf{G} \tag{2.28}$$

This system has indeed a unique solution because the matrix $\mathbf{K}^T\mathbf{K}+\alpha^2\mathbf{I}$ is positive definite and thus non-singular. To prove that this solution is Tikhonov solution, we substitute $\mathbf{K} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ and utilize that $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices

$$\mathbf{F} = \mathbf{V}(\mathbf{S}^T\mathbf{S} + \alpha^2\mathbf{I})^{-1}\mathbf{S}^T\mathbf{U}^T\mathbf{G} \tag{2.29}$$

Then knowing that $(\mathbf{S}^T\mathbf{S} + \alpha^2\mathbf{I})^{-1}\mathbf{S}^T = \mathrm{diag}[s_1/(s_1+\alpha^2), ..., s_r/(s_r+\alpha^2), 0, ...]$, it can be written as

$$\mathbf{F} = \sum_{j=1}^{r} \frac{s_j}{s_j + \alpha^2} \ (\mathbf{u_j}^T \mathbf{G}) \ \mathbf{v_j} \ , \tag{2.30}$$

which is nothing but Eq. (2.26).

It is worth noting that Eq. (2.28) is the normal equations of a least squares problem with modified matrix and data

$$\min_{\mathbf{F}\in\mathbb{R}^n} \left\| \begin{pmatrix} \mathbf{K} \\ \alpha\mathbf{I} \end{pmatrix} \mathbf{F} - \begin{pmatrix} \mathbf{G} \\ \emptyset \end{pmatrix} \right\|_2 , \tag{2.31}$$

which is also equivalent to the minimization problem

$$\min_{\mathbf{F}\in\mathbb{R}^n} \|\mathbf{K}\ \mathbf{F} - \mathbf{G}\|_2^2 + \alpha^2 \|\mathbf{F}\|_2^2 \tag{2.32}$$

leading to Eq. (2.28). This formulation allows us to interpret Tikhonov solution as the one that balances between the residual and the model norm, and this balance is controlled by the regularization parameter α.

When α is very small, we approach the least squares solution which fits the data very well but has a very large norm. When α is very large, then the solution has a small norm but fits the data badly, and thus has a large residual. This motivates the L-curve method [7] for choosing the best value of α. The L-curve method suggests plotting the model norm versus the residual norm on a log-log scale for different values of α. The curve will have an L shape and the value of α at the corner of the L is taken as the best value.

Fig. 2.6 shows Tikhonov regularized solutions of our test case for different values of α while in Fig. 2.7 we show the L-curve which suggests that the best compromise is achieved for $\alpha = 10^{-4}$.

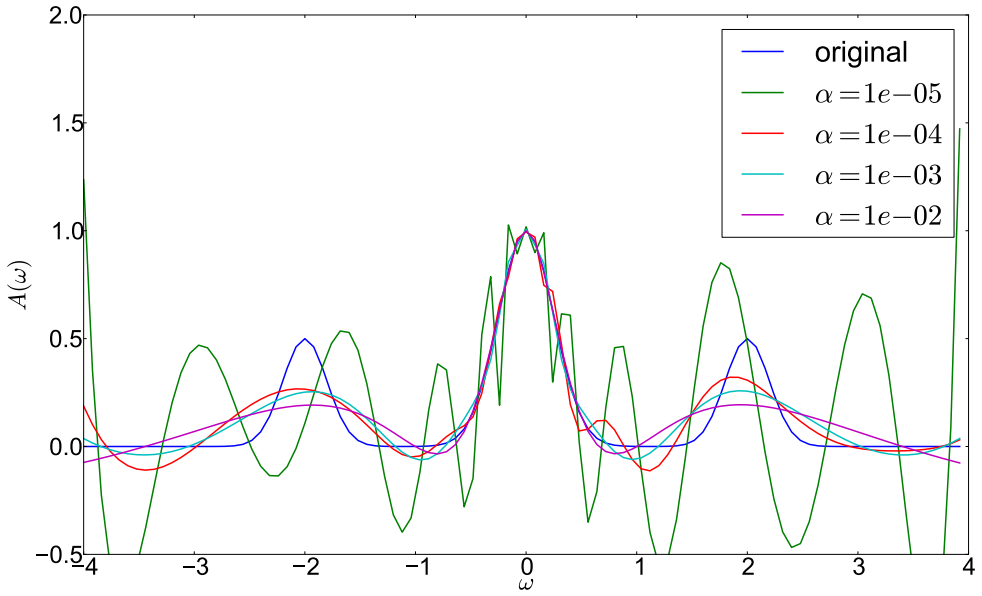

Fig. 2.6.: Tikhonov solutions for different values of parameter α.

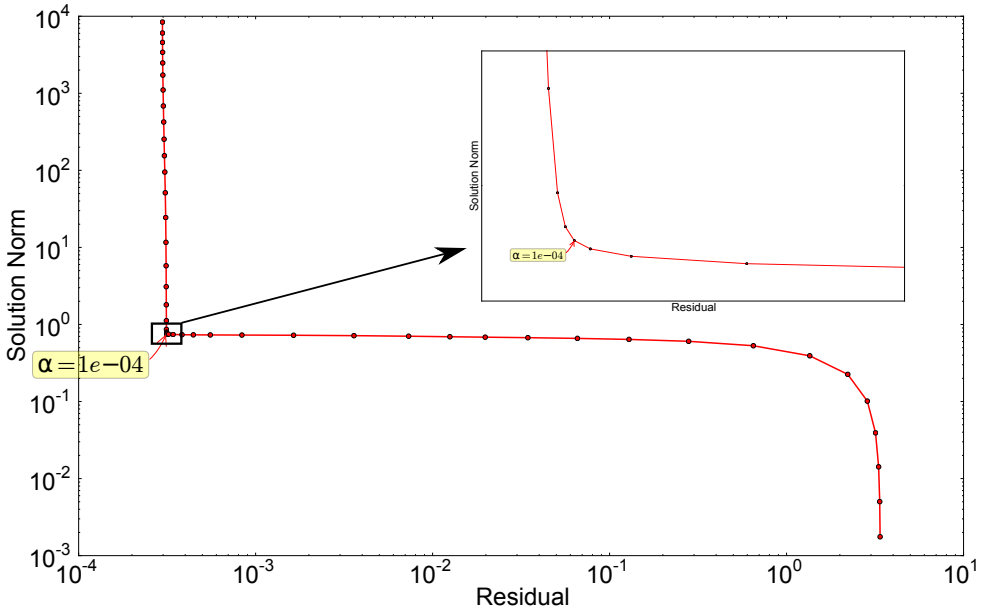

Fig. 2.7.: L-curve for the example test case. The best value of α is the one corresponding to the corner. It equals 10^{-4}.

2.6. Non-negativity Constraint

The previous methods are general and apply to all Fredholm integral problems of first kind. Now we utilize a simple, yet important, piece of knowledge about the analytic continuation problem; *The model is non-negative.* So we look for the *non-negative least squares solution (NNLS)*

$$\mathbf{F}_{\mathrm{NNLS}} = \underset{\mathbf{F}\in\mathbb{R}^n, \mathbf{F}\geq 0}{\arg\min} \parallel \mathbf{K}\,\mathbf{F} - \mathbf{G} \parallel_2 \tag{2.33}$$

Fig. 2.8 shows a simple illustration of the difference between the least squares solution and the non-negative least squares solution for a two-dimensional case. Of course, the two solutions can in principle be the same but it is highly unlikely. Note how the NNLS solution lies on the boundary of the positive region which means that it will probably have many zeros in the multidimensional case.

In Fig. 2.9 we show both the least squares solution and the non-negative least squares solution for our test case. While the LS solution is completely useless, the NNLS solution, although still noisy, captures some of the model's structure and is a great improvement over the LS one. Ref. [8] describes an algorithm to obtain the NNLS solution. It is an iterative algorithm that starts from the zero model. Then it modifies the model such that the residual is reduced at each step making sure to maintain the constraint. The algorithm is proved to converge in a finite number of steps.

We can combine the non-negativity and regularization to get a non-negative Tikhonov solution. Remembering Eq. (2.31), the non-negative Tikhonov (NNT) solution is nothing but the NNLS solution of a modified problem; data is padded with zeros and the matrix $\mathbf{K}$ is padded with a multiple of unity.

In Fig. 2.10 we show the non-negative Tikhonov solutions of the test case for different values of α and in Fig. 2.11 we show the L-curve used with NNT. Notice that unlike for the normal Tikhonov (see Fig. 2.7) the solution norm has a maximum value for very small α because the constraint already prevents solutions of very large norm.

2.7. Bayesian Approach

In the previous section we utilized our knowledge about the model, while in this section we will utilize our knowledge about the noise on the data. Let $\mathbf{G}^\star$ be the unknown exact data and let $\tilde{\mathbf{G}}$ be the actual noisy data we have to work with. We assume that computation introduces noise that is normally distributed and has a zero mean.[3] The noise on different data components could be independent or correlated. More generally, the noisy data will be distributed around the exact data as a multivariate normal distribution

$$\tilde{\mathbf{G}} \sim \mathcal{N}(\mathbf{G}^\star, \mathbf{C}) \tag{2.34}$$

[3]This assumption is justified by the central limit theorem.

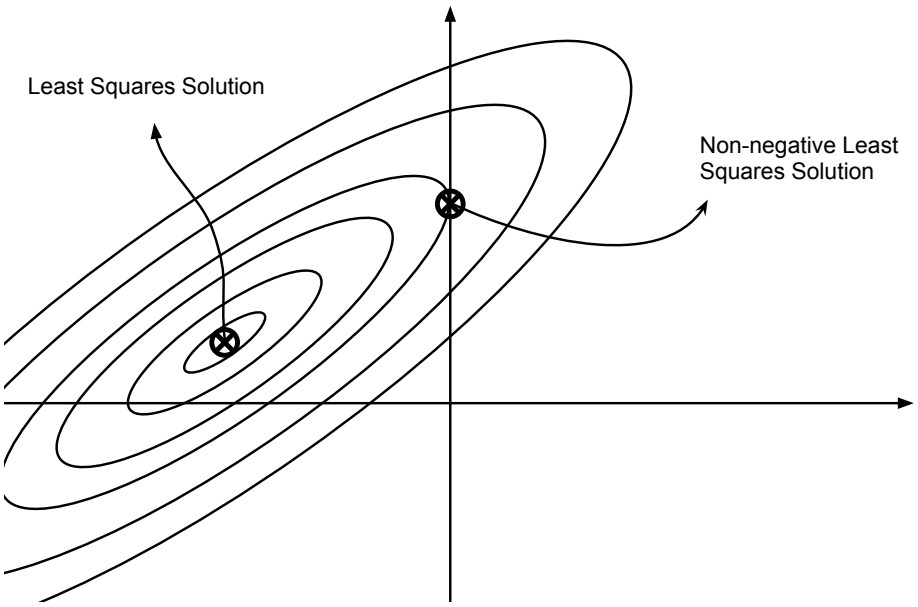

Fig. 2.8.: An illustration of the difference between the least squares solution and the non-negative least squares solution for a two-dimensional case. The ellipses represent the contour of the function that least squares methods try to minimize (the residual).

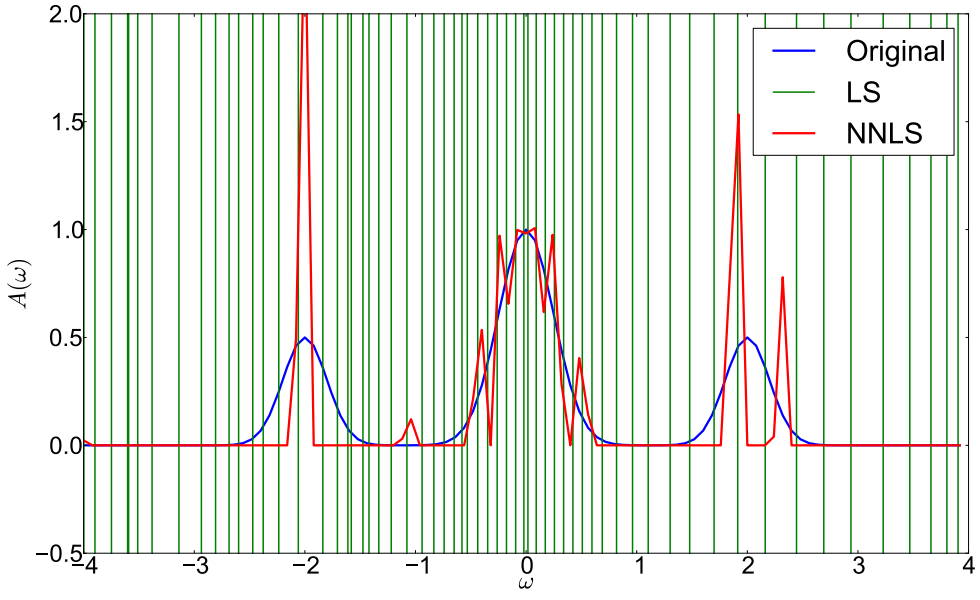

Fig. 2.9.: A comparison of the least squares solution (LS) and the non-negative least squares solution (NNLS) for the test case.

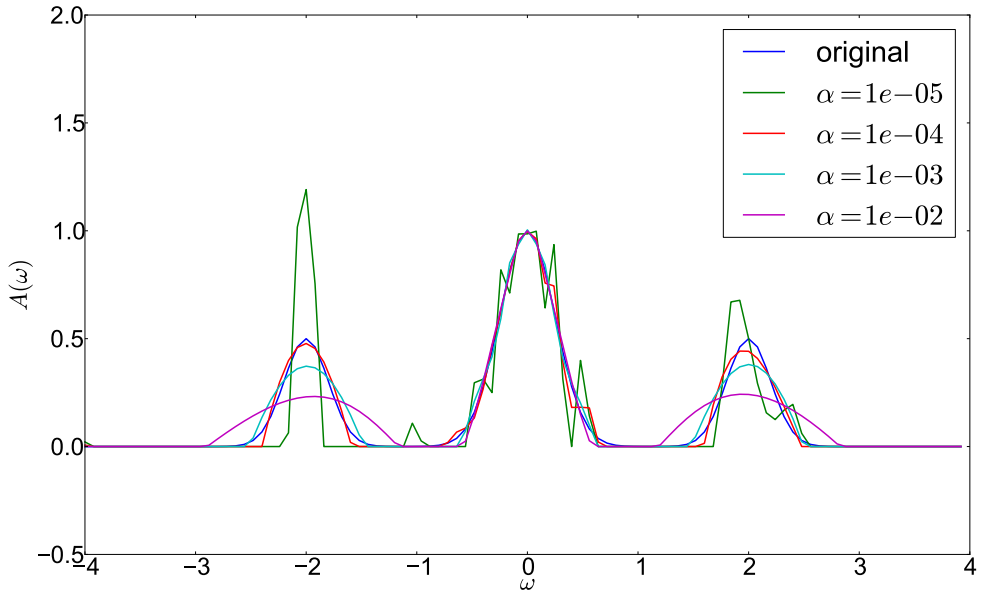

Fig. 2.10.: Non-negative Tikhonov solutions for different values of parameter α.

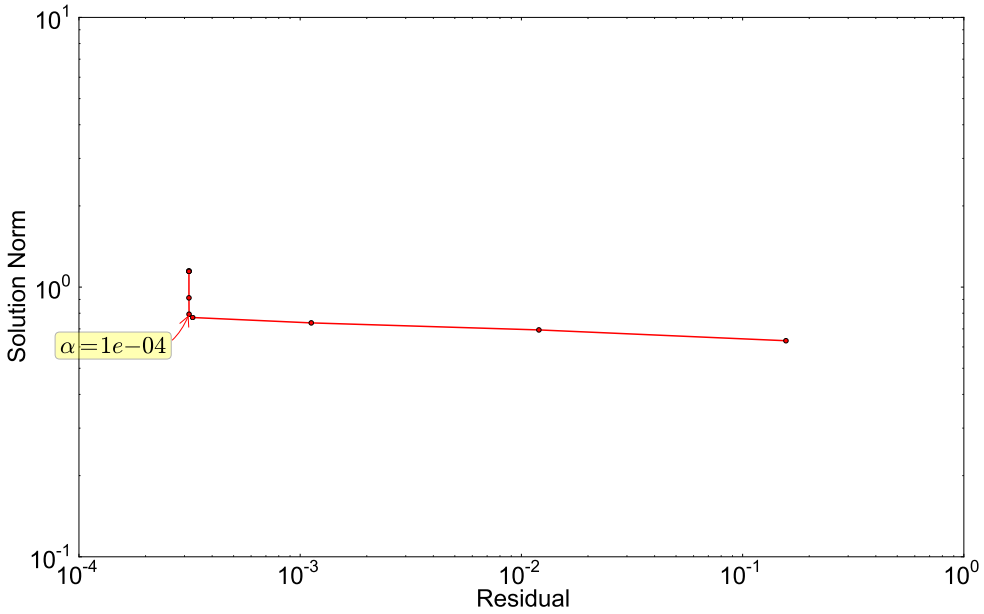

Fig. 2.11.: L-curve for the example test case with non-negative Tikhonov method. The best value of α is the one corresponding to the corner and it equals 10^{-4}. Compare with Fig. 2.9 and notice how the Tikhonov method gives smooth solutions compared to least squares methods.

where the coavraince matrix[4] $\mathbf{C}$ can be estimated from multiple independent data samples.[5]

Using Bayesian inference, we can translate the probability distribution of the data into a probability distribution for the models

$$P\left[\mathbf{F}|\tilde{\mathbf{G}}\right] = \frac{P\left[\tilde{\mathbf{G}}|\mathbf{G}^{\star} = \mathbf{KF}\right]\ P\left[\mathbf{F}\right]}{P[\tilde{\mathbf{G}}]} \tag{2.35}$$

The previous relation can be read as following: The probability that model $\mathbf{F}$ is the actual model given the measured data $\tilde{\mathbf{G}}$, equals the probability of measuring $\tilde{\mathbf{G}}$ given that $\mathbf{KF}$ is the exact data, multiplied by the prior probability of $\mathbf{F}$ being the actual model, and divided by the prior probability of measuring data $\tilde{\mathbf{G}}$.

Equipped with a probability distribution of the models, we can choose the model with the maximum probability as "the best" solution, and so the problem is to find the model maximizing Eq. (2.35). We ignore $P[\tilde{\mathbf{G}}]$ because it is a constant independent of the model. Assuming for the moment that we have no prior knowledge about the model, then $P\left[\mathbf{F}\right]$ is also a constant and can be ignored in the maximization process. The only term left is $P\left[\tilde{\mathbf{G}}|\mathbf{G}^{\star} = \mathbf{KF}\right]$ which is a multivariate normal distribution as discussed above. Putting things together we have

$$P\left[\mathbf{F}|\tilde{\mathbf{G}}\right] \propto \exp\left\{-\frac{1}{2}(\mathbf{KF} - \tilde{\mathbf{G}})^T\mathbf{C}^{-1}(\mathbf{KF} - \tilde{\mathbf{G}})\right\} \tag{2.36}$$

Maximizing $P\left[\mathbf{F}|\tilde{\mathbf{G}}\right]$ is equivalent to minimizing $(\mathbf{KF} - \tilde{\mathbf{G}})^T\mathbf{C}^{-1}(\mathbf{KF} - \tilde{\mathbf{G}})$, because, the exponentially decaying function is a monotonically decreasing function. Since $\mathbf{C}$ is a covaraince matrix, its inverse $\mathbf{C}^{-1}$ is symmetric and positive-definite[6] so it can be Cholesky decomposed into $\mathbf{C}^{-1} = \mathbf{W}^T\mathbf{W}$ and the problem becomes

$$\min_{\mathbf{F}\in\mathbb{R}^n} \parallel \mathbf{WKF} - \mathbf{W}\tilde{\mathbf{G}} \parallel_2^2 \tag{2.37}$$

This nothing but a least squares problem with a modified matrix $\mathbf{WK}$ and modified data $\mathbf{W}\tilde{\mathbf{G}}$. Indeed solving this modified problem is called *Generalized Least Squares* and when the noise on different data components are identical and uncorrelated, then the generalized least squares solution is the same as the usual least squares solution.

In our test case, the covariance matrix is diagonal, with diagonal element i equals $10^{-4} \times \tilde{\mathbf{G}}_{\mathbf{i}}$. In Fig. 2.12, we show the generalized least squares solution which is not much different from the normal least squares solution (see Fig. 2.2). This is something to be expected because there are no correlations between the data components and the differences between their noise variances is small (see Fig. 2.13), so $\mathbf{W}$ hardly modifies the data and the matrix.

[4]The covariance matrix is usually detonated as $\mathbf{\Sigma}$ but we denote it here as $\mathbf{C}$ to avoid confusion with the self-energy used in many-body physics.

[5]Samples from Monte Carlo simulations are correlated but there are techniques to remove this correlation.

[6]Because $\mathbf{C}$ is symmetric and positive-definite.

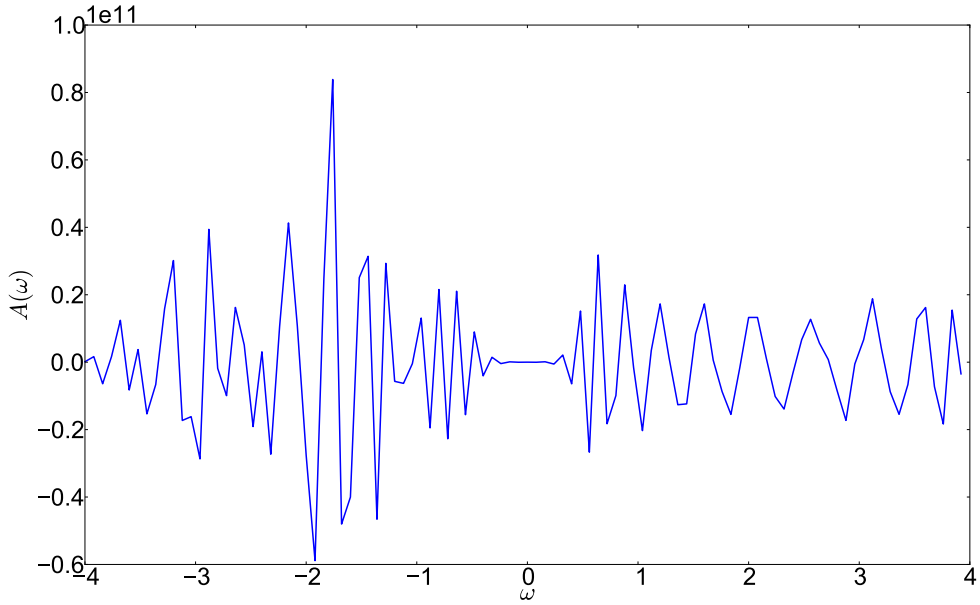

Fig. 2.12.: The spectral function reconstructed using generalized least squares method. The solution is not better than the normal least squares one (Fig. 2.2).

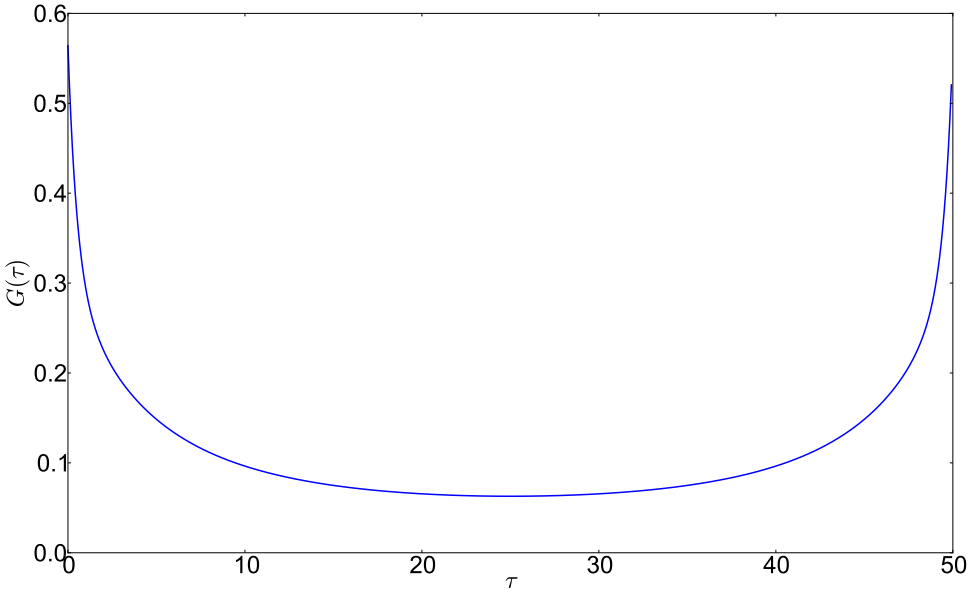

Fig. 2.13.: The data for the test case. Note that all the values are within the same order of magnitude so the noise on each one of them is also of the same order of magnitude which explains why for our test case the generalized least squares solution is not much different form the usual least squares one.

2.7.1. Smoothness as prior knowledge – Tikhonov Revisited

What do we assume about the model that makes the least squares solutions (e.g. Fig. 2.2 and Fig. 2.12) unacceptable?
We assume that the values of the model should have reasonable magnitudes, and we also assume some kind of smoothness; both of which, the aforementioned solutions clearly do not satisfy. Those assumptions can be formulated mathematically, for example, by the following prior distribution of models

$$P\left[\mathbf{F}\right] \propto \exp\left\{-\frac{1}{2}\alpha^2 \parallel \mathbf{F} \parallel_2^2\right\} . \tag{2.38}$$

Combining it with the likehood $P\left[\tilde{\mathbf{G}}|\mathbf{G}^\star = \mathbf{KF}\right]$, we get the desired distribution

$$P\left[\mathbf{F}|\tilde{\mathbf{G}}\right] \propto \exp\left\{-\frac{1}{2}(\mathbf{KF}-\tilde{\mathbf{G}})^T\mathbf{C}^{-1}(\mathbf{KF}-\tilde{\mathbf{G}}) - \frac{1}{2}\alpha^2 \parallel \mathbf{F} \parallel_2^2\right\} . \tag{2.39}$$

If we follow the same steps as in the previous section, we find that the model with maximum posterior probability $P\left[\mathbf{F}|\tilde{\mathbf{G}}\right]$ is the one that solves the problem

$$\min_{\mathbf{F}\in\mathbb{R}^n} \parallel \mathbf{WKF} - \mathbf{W}\tilde{\mathbf{G}} \parallel_2^2 + \alpha^2 \parallel \mathbf{F} \parallel_2^2 \tag{2.40}$$

Comparing it with Eq. (2.32), we see that this is the Tikhonov solution for a modified matrix $\mathbf{WK}$ and modified data $\mathbf{W}\tilde{\mathbf{G}}$.

2.7.2. Non-negativity as prior knowledge

We know for sure that values of the model in analytic continuation problems are non-negative. This can be expressed using the following prior distribution

$$P\left[\mathbf{F}\right] = \begin{cases} constant & \text{for } \mathbf{F} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.41}$$

It should be clear at this point that if we try to find the model that maximizes $P\left[\mathbf{F}|\tilde{\mathbf{G}}\right]$, we will arrive to the non-negative least squares solution with a modified matrix and modified data (Fig. 2.8).

2.8. Stochastic Sampling Methods

There is something different about the non-negativity that sets its probability distribution apart from the previous ones. *It is skewed!* The probability distribution without prior

knowledge and the one with smoothness prior are both multivariate normal distributions and so their mean is the same as the modal (i.e. the model with the maximum probability), and they are indeed the best representatives of their corresponding distributions. However, the probability distribution with non-negativity prior is a *truncated* multivariate normal which can be a highly-skewed distribution and *thus the model with the highest probability is not the best estimate but rather the mean.* This is the motivation behind stochastic sampling methods which try to compute the mean

$$\overline{\mathbf{F}} = \frac{1}{C'} \int\limits_{\mathbf{F}\geq 0} d\mathbf{F}\ \mathbf{F}\ \exp\left\{-\frac{1}{2}(\mathbf{KF}-\tilde{\mathbf{G}})^T\mathbf{C}^{-1}(\mathbf{KF}-\tilde{\mathbf{G}})\right\} \tag{2.42}$$

where C' is a normalization constant, we do not need in practice.

Computing the mean $\overline{\mathbf{F}}$ is done by sampling the space of models using some Monte-Carlo method and then averaging the samples. A common approach is to use the Metropolis algorithm to do the sampling as in Refs. [9, 10, 11]. They start from some initial non-negative model $\mathbf{F}'$ and make changes to its components to obtain the next sample $\mathbf{F}''$. The new sample is then accepted or rejected according to some criteria. Although the details of the sampling may differ between the different approaches [9, 10, 11], they share the following features:

- New samples are obtained by manipulating the model components directly.
- Non-negativity is imposed easily on different components by suggesting only changes that preserve the constraint.
- Once a model with a high probability is found, it takes a considerable number of steps to find a different model with high probability. This is because of the high correlation between the different components.
- To avoid being stuck around a specific model of high probability, a simulated annealing procedure with fictitious temperature parameter is used.

2.8.1. Stochastic Mode Sampling (SMS) - Theory

In this section, we present a sampling approach that avoids the correlation between the model's components and thus leads to more efficient sampling.

Since $\mathbf{C}$ is a covaraince matrix, its inverse $\mathbf{C}^{-1}$ is symmetric and positive-definite so it can be decomposed using Cholesky decomposition into $\mathbf{C}^{-1} = \mathbf{W}^T\mathbf{W}$, and the exponent of the Gaussian weight in Eq. (2.42) can be rewritten as

$$\chi^2[\mathbf{F}] \coloneqq (\mathbf{KF}-\tilde{\mathbf{G}})^T\mathbf{C}^{-1}(\mathbf{KF}-\tilde{\mathbf{G}}) = (\mathbf{WKF}-\mathbf{W}\tilde{\mathbf{G}})^T(\mathbf{WKF}-\mathbf{W}\tilde{\mathbf{G}})\ . \tag{2.43}$$

Taking the Singular Value Decomposition (SVD) of the modified matrix $\mathbf{WK} = \mathbf{USV}^T$, we get

$$\chi^2[\mathbf{F}] = (\mathbf{SV}^T\mathbf{F}-\mathbf{U}^T\mathbf{W}\tilde{\mathbf{G}})^T(\mathbf{SV}^T\mathbf{F}-\mathbf{U}^T\mathbf{W}\tilde{\mathbf{G}}) \tag{2.44}$$

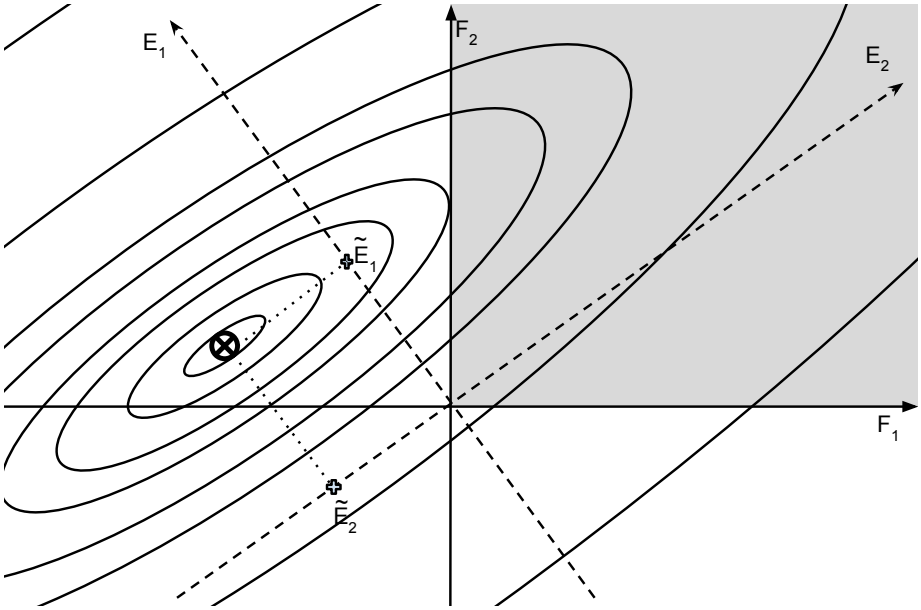

Fig. 2.14.: 2D Illustration of the difference between typical stochastic sampling methods and the new stochastic modes sampling. The ellipses represent the contours of the probability distribution in models' space. Eq. (2.42) expresses this distribution in terms of $\mathbf{F1}, \mathbf{F2}$ while Eq. (2.55) expresses it in terms of $\mathbf{E1}, \mathbf{E2}$. The shaded region represent the region of models' space that should be sampled. Although this region is the same for both cases, it is complex to express in terms of $\mathbf{E1}, \mathbf{E2}$, while it is simply half open intervals in terms of $\mathbf{F1}, \mathbf{F2}$.

where $\mathbf{U}\mathbf{U}^{\mathbf{T}} = \mathbf{I}$ is used.

We denote the projection of the model on the modes (aka right singular vectors) as

$$\mathbf{E} \coloneqq \mathbf{V}^T\mathbf{F}, \tag{2.45}$$

while we denote the projection of the modified data on the left singular vectors as

$$\tilde{\mathbf{H}} \coloneqq \mathbf{U}^T\mathbf{W}\tilde{\mathbf{G}} \,. \tag{2.46}$$

Using the previous notation, we write

$$\bar{\chi}^2[\mathbf{E}] \coloneqq (\mathbf{SE} - \tilde{\mathbf{H}})^T(\mathbf{SE} - \tilde{\mathbf{H}}) = \chi^2[\mathbf{F}] \,. \tag{2.47}$$

We can change the integration variable in Eq. (2.42) from $\mathbf{F}$ to $\mathbf{E}$ without further changes, because they are related by an orthogonal transformation and the determinant of the Jacobian is one

$$\overline{\mathbf{F}} = \frac{1}{C'} \int\limits_{(\mathbf{VE})\geq 0} d\mathbf{E}\ \mathbf{VE} \exp\left\{-\frac{1}{2}\bar{\chi}^2[\mathbf{E}]\right\} = \frac{\mathbf{V}}{C'} \int\limits_{(\mathbf{VE})\geq 0} d\mathbf{E}\ \mathbf{E} \exp\left\{-\frac{1}{2}\bar{\chi}^2[\mathbf{E}]\right\} \tag{2.48}$$

where the last equality holds due to the linearity of integration.

Let r be the number of non-zero singular values, then we can decompose $\tilde{\chi}^2$ as following

$$\bar{\chi}^2[\mathbf{E}] = \sum_{j=1}^{m}(s_j\mathbf{E_j} - \tilde{\mathbf{H}}_\mathbf{j})^2 = \sum_{j=1}^{r} s_j^2(\mathbf{E_j} - \tilde{\mathbf{H}}_\mathbf{j}/s_j)^2 + \sum_{j=r+1}^{m}(\tilde{\mathbf{H}}_\mathbf{j})^2 \tag{2.49}$$

$$= \sum_{j=1}^{r} s_j^2(\mathbf{E_j} - \tilde{\mathbf{E}}_\mathbf{j})^2 + \text{Residual} \tag{2.50}$$

$$\tag{2.51}$$

where

$$\tilde{\mathbf{E}}_\mathbf{j} \coloneqq \tilde{\mathbf{H}}_\mathbf{j}/s_j \tag{2.52}$$

$$\text{Residual} \coloneqq \begin{cases} \sum_{j=r+1}^{m}(\tilde{\mathbf{H}}_\mathbf{j})^2 & \text{for } r < m \\ 0 & \text{for } r \geq m \end{cases} . \tag{2.53}$$

Going back to the vector form, we have

$$\bar{\chi}^2[\mathbf{E}] = (\mathbf{E} - \tilde{\mathbf{E}})^T\mathbf{S}^T\mathbf{S}(\mathbf{E} - \tilde{\mathbf{E}}) + \text{Residual} . \tag{2.54}$$

By substituting the last relation in Eq. (2.48) and absorbing the constant $\exp\{-\text{Residual}/2\}$ in the normalization factor, we get the desired relation

$$\overline{\mathbf{F}} = \frac{\mathbf{V}}{C''} \int\limits_{(\mathbf{VE})\geq 0} d\mathbf{E}\, \exp\left\{-\frac{1}{2}(\mathbf{E} - \tilde{\mathbf{E}})^T\mathbf{S}^T\mathbf{S}(\mathbf{E} - \tilde{\mathbf{E}})\right\} \cdot \mathbf{E} \tag{2.55}$$

This equation shows that we can compute $\overline{\mathbf{F}}$ by sampling the coefficients $\mathbf{E}$, resulting from projecting the model on the modes, instead of sampling the components of the model directly. Actually, Eq. (2.55) and Eq. (2.42) express the same probability distribution (a truncated multivariate normal distribution) over the models but in two different bases . The advantage of Eq. (2.55) over Eq. (2.42) is that the coefficients $\mathbf{E_i}$ are uncorrelated because the matrix $(\mathbf{S}^T\mathbf{S})$ is diagonal, while the $\mathbf{F_i}$ are highly correlated because of the large non-diagonal elements of $\mathbf{K}^T\mathbf{C}^{-1}\mathbf{K}$. The disadvantage is, however, that non-negativity constraint is harder to express in terms of $\mathbf{E_i}$, while it consists simply of half-open intervals in terms of the $\mathbf{F_i}$ (see Fig. 2.14).

2.8.2. Stochastic Mode Sampling (SMS) - Algorithm

Without the non-negativity constraint, the coefficient $\mathbf{E_i}$ is completely independent of other coefficients and it is distributed as a normal random variable of mean $\mu_i = \tilde{\mathbf{E}}_\mathbf{i}$ and variance $\sigma_i^2 = 1/s_i^2$. Since the singular values are sorted descendingly, we have more information about the leading modes than about later ones. The extreme case is when the singular value is zero,[7] which happens for $i > r$. In this case, we have no information about the

[7] up to numerical accuracy

mode from the data (we call such modes *free modes*) and the corresponding coefficient is distributed uniformly.[8]

Taking the constraint into account, the conditional distribution of coefficient $\mathbf{E_i}$ is a normal distribution with mean $\mu_i = \tilde{\mathbf{E}}_\mathbf{i}$ and variance $\sigma_i^2 = 1/s_i^2$ but truncated to the interval $[a, b]$ where the model is non-negative (Ref. [15] proves that the conditional probability of a truncated multivariate normal distribution is again a multivariate normal distribution). The limits a, b depend on the values of all other coefficients and they can be expressed in terms of them as following

$$\mathbf{VE} \geq 0 \Rightarrow \sum_{j=1}^{n} \mathbf{v_j E_j} \geq 0 \Rightarrow \mathbf{v_i E_i} + \underbrace{\sum_{j=1, j\neq i}^{n} \mathbf{v_j E_j}}_{:=c} \geq 0 \Rightarrow \mathbf{v_i E_i} \geq c \tag{2.56}$$

$$\Rightarrow \begin{cases} \mathbf{E_i} \geq c/\mathbf{v_i}[k] \text{ if } \mathbf{v_i}[k] > 0 \\ \mathbf{E_i} \leq c/\mathbf{v_i}[k] \text{ if } \mathbf{v_i}[k] < 0 \end{cases} \tag{2.57}$$

$$\Rightarrow \begin{cases} a = \max & \{c/\mathbf{v_i}[k] : \mathbf{v_i}[k] > 0\} \cup \{-\infty\} \\ b = \min & \{c/\mathbf{v_i}[k] : \mathbf{v_i}[k] < 0\} \cup \{+\infty\} \end{cases} . \tag{2.58}$$

If $a > b$ then the interval is empty and for the specified values of $\mathbf{E_{j\neq i}}$, there is no allowed value of $\mathbf{E_i}$, but this never happens because, as explained below, we start from an allowed model and ensure the constraint at each step. For the case of free modes (i.e. $s_i = 0$), we have a uniform distribution in the interval $[a, b]$.

Now we turn to the problem of sampling Eq. (2.55). We use a variant of the Metropolis algorithm called *Gibbs Sampling* [12, 13]. Gibbs sampling is useful for sampling a joint probability distribution when the conditional ones are known and easy to sample. It starts from some initial sample $\mathbf{E}^{(0)}$. Then every new sample $\mathbf{E}^{(k)}$ is generated from the previous one $\mathbf{E}^{(k-1)}$ by sampling each coefficient conditional on the value of all other coefficients where the value of a coefficient is updated as soon as it is sampled. More precisely, $\mathbf{E_i}^{(k)}$ is sampled from the conditional probability $P[\mathbf{E_i}|\mathbf{E_1}^{(k)}, ..., \mathbf{E_{i-1}}^{(k)}, \mathbf{E_{i+1}}^{(k-1)}, ..., \mathbf{E_n}^{(k-1)}]$. Gibbs sampling is applicable to our case because the conditional distributions are either uniform distributions or truncated normal distributions with computable parameters (see Appendix B for sampling a truncated univariate normal distribution). Since each coefficients $\mathbf{E_i}$ is sampled such that the model is non-negative, the constraint is satisfied at every step provided we start from an allowed model. The starting model could be any non-negative model like all zeros, the non-negative least squares solution, or even a random non-negative model.

In the listing next page, we provide the pseudo code of the stochastic mode sampling method. It takes four parameters: The matrix $\mathbf{K}$ resulting from discretizing the kernel, the data $\tilde{\mathbf{G}}$ which may be the average of several data samples, the covariance matrix of the data $\mathbf{C}$ which maybe estimated from those data samples, and the desired number of model samples. The procedure returns samples from the models' space according to the previously-discussed distribution.

[8] This is justifiable by the fact that the normal distribution approaches the uniform one as the variance tends to infinity.

Stochastic Mode Sampling

procedure SMS($\mathbf{K}, \tilde{\mathbf{G}}, \mathbf{C}$, samplesNum)
  $m, n \leftarrow \text{shape}(\mathbf{K})$
  $p \leftarrow \min(m, n)$
  $\mathbf{W}^T\mathbf{W} \leftarrow \text{cholesky}(\mathbf{C}^{-1})$
  $\mathbf{U}, \mathbf{S}, \mathbf{V}^T \leftarrow \text{svd}(\mathbf{WK})$
  $\mathbf{s} \leftarrow \text{diagonal}(\mathbf{S})$ ▷ retrieve singular values
  $r \leftarrow$ number of non-zero singular values ▷ up to numerical accuracy
  $\tilde{\mathbf{H}} \leftarrow \mathbf{U^T}\tilde{\mathbf{G}}$
  $\tilde{\mathbf{E}} \leftarrow \tilde{\mathbf{H}}/\mathbf{s}$ ▷ element-wise division
  $\sigma \leftarrow 1/\mathbf{s}$ ▷ element-wise reciprocal
  $\mathbf{F} \leftarrow$ some non-negative model ▷ e.g. zeros or NNLS or random
  $\mathbf{E} \leftarrow \mathbf{V^T F}$
  samples $\leftarrow \{\}$
  while samplesNum > 0 **do**
    for i=1 **to** n **do**
      $a \leftarrow -\infty$
      $b \leftarrow +\infty$
      for k=1 **to** n **do**
        $d \leftarrow \mathbf{E_i} - \mathbf{F_k}/\mathbf{V_{k,i}}$
        If $\mathbf{V_{k,i}} > 0$ **and** $d > a$ **then** $a \leftarrow d$
        If $\mathbf{V_{k,i}} < 0$ **and** $d < b$ **then** $b \leftarrow d$
      end for
      if $i \leq r$ **then**
        $e \leftarrow \text{truncNorm}(\tilde{\mathbf{E}}_\mathbf{i}, \sigma_\mathbf{i}, a, b)$ ▷ truncated normal random variable
      else
        $e \leftarrow \text{uniform}(a, b)$ ▷ uniform random variable
      end if
      $\mathbf{F} \leftarrow \mathbf{F} + (e - \mathbf{E_i})\mathbf{V}_{-,\mathbf{i}}$
      $\mathbf{E_i} \leftarrow e$
    end for
    samplesNum $\leftarrow$ samplesNum $- 1$
    samples $\leftarrow$ samples $\cup\{\mathbf{F}\}$
  end while
  return samples
end procedure

The final step is now estimating the distribution mean $\overline{\mathbf{F}}$ from the finite set of samples and, desirably, error bars of this estimation. Gibbs sampling, like other Mote Carlo methods, produces *correlated samples*. As far as the mean is concerned, this is not a problem. The mean is simply the average of the samples assuming we neglect a sufficient number of samples at the beginning (called burn-in period which is usually 1000 to 5000 samples) to grantee that the Markov chain has reached its stationary distribution. Estimating the error bars , however, is tricky to be done with correlated samples and there are different methods to achieve it. We use the blocking method discussed in Appendix C.

Computational Complexity In the initialization phase, we perform three costly operations. The computational complexity of the inversion and Cholesky decomposition of a general covariance matrix is $\mathcal{O}(m^3)$. However, if the data components are uncorrelated then the coveriance matrix is diagonal and both operations are $\mathcal{O}(m)$. The singular value decomposition complexity is $\mathcal{O}(n^3)$, assuming m is of the same order of n. For each sample, we loop over all modes (there are n modes), and for each mode, we have $\mathcal{O}(n)$ operations to ensure the constraint; In total, we have $\mathcal{O}(n^2)$ operation per sample. Since almost always $samplesNum \gg n$, the total complexity of the method is $\mathcal{O}(samplesNum \times n^2)$.

Parallelization SMS is embarrassingly parallel like most Monte Carlo methods. Several instances of the algorithm with different starting models can be run in parallel and the samples generated by one instance are uncorrelated to the samples generated by another instance. This provides us with a coarse-grained parallelization scheme, but there is yet a fine-grained one. The code for finding the interval limits a, b (lines 15-22) can be parallelized by evaluating d for different k in parallel and then finding the conditional min and max. Also the code for updating the model (line 28) can be parallelized easily. So we can apply two levels of parallelism to SMS. In case we have access to several multicore processors, we can apply the coarse-grained level by mapping different instances to different processors (using MPI for example) while applying the the fine-grained level inside each processor (using OpenMP for example). In case we have access to a GPGPU (General-Purpose Graphics Processing Unit), we can apply the coarse-grained level by mapping different instances to different workgroups while applying the fined-grained level inside each workgroup. This can be done using either OpenCL or CUDA.

Results Fig. 2.15 shows the application of the SMS method to our test case. Since the noise on the data is relative uncorrelated noise of standard deviation $\sigma = 10^{-4}$, the covariance matrix $\mathbf{C}$ is a diagonal matrix whose diagonal element $\mathbf{C}_{j,j}$ is $(\tilde{\mathbf{G}}_j\sigma)^2$. The solution is the average of $2^{24} \approx 16 \times 10^6$ samples generated after a burn-in period of 10^4 samples. Notice that external peaks are sharper than they should be, while the middle peak gets some spurious features.

In Fig. 2.16, we compare the stochastic modes sampling (SMS) solution with the non-negative least squares (NNLS) solution obtained using the modified matrix $\mathbf{WK}$ and modified data $\mathbf{W}\tilde{\mathbf{G}}$. Clearly, the SMS solution is better than the NNLS solution. As discussed earlier, the NNLS solution represents the solution with the maximum probability (the

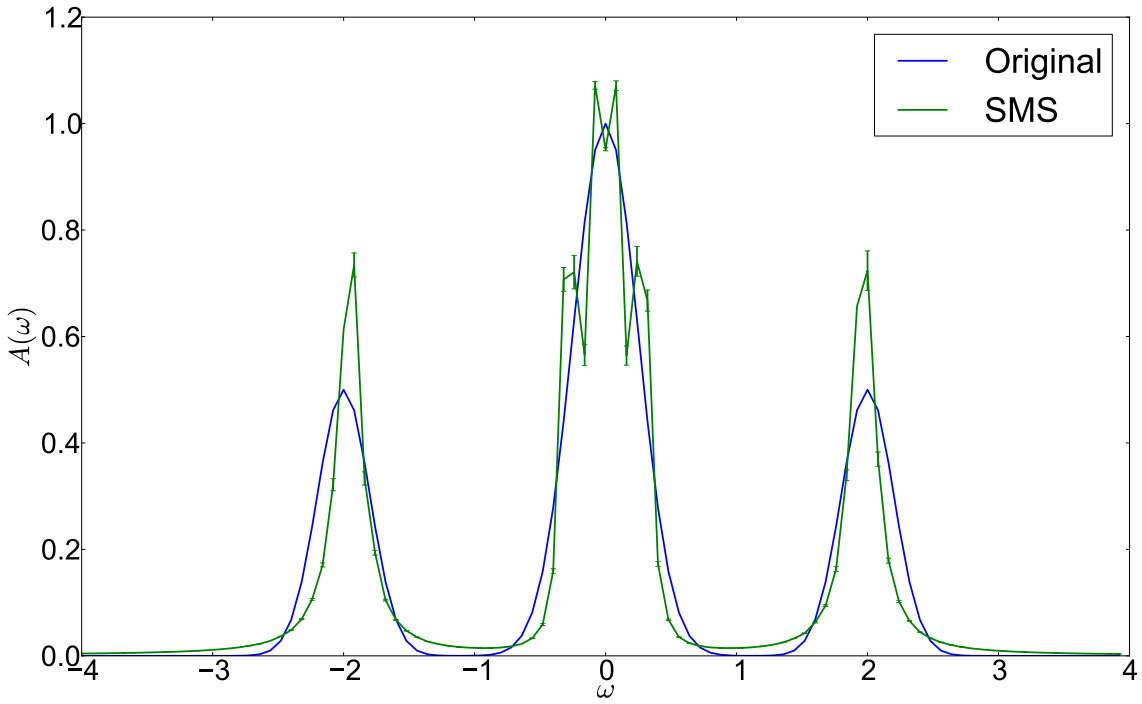

Fig. 2.15.: Stochastic Mode Sampling Solution

modal) while the SMS solution represents the mean. Since the probability distribution is skewed, its modal (the NNLS solution) lies on the border of the non-negative area and thus contains many zeros.

In Fig. 2.17, we compare the stochastic modes sampling (SMS) solution with the non-negative Tikhonov (NNT) solution with $\alpha = 10^{-4}$. On the one hand, NNT is much faster than SMS and provides smoother solution with fewer spurious features. On the other hand, SMS gives a reasonably good solution using only information we are certain about, namely the non-negativity and the covariance matrix. In addition, the "smoothness" of SMS comes as a result averaging while the "smoothness" of NNT is imposed by hand and depends on a parameter α that needs to be tuned heuristically. Finally, note that NNT solution is clamped outside the peaks instead of approaching the axis smoothly.

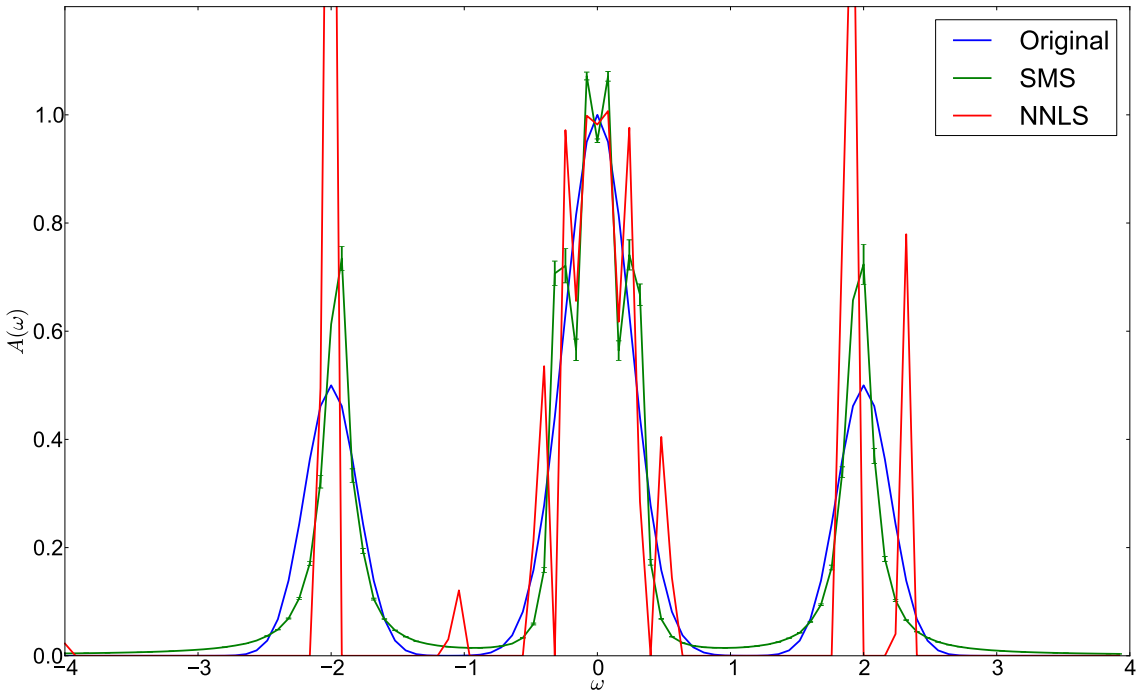

Fig. 2.16.: Stochastic mode sampling (SMS) solution vs. non-negative least squares (NNLS) solution.

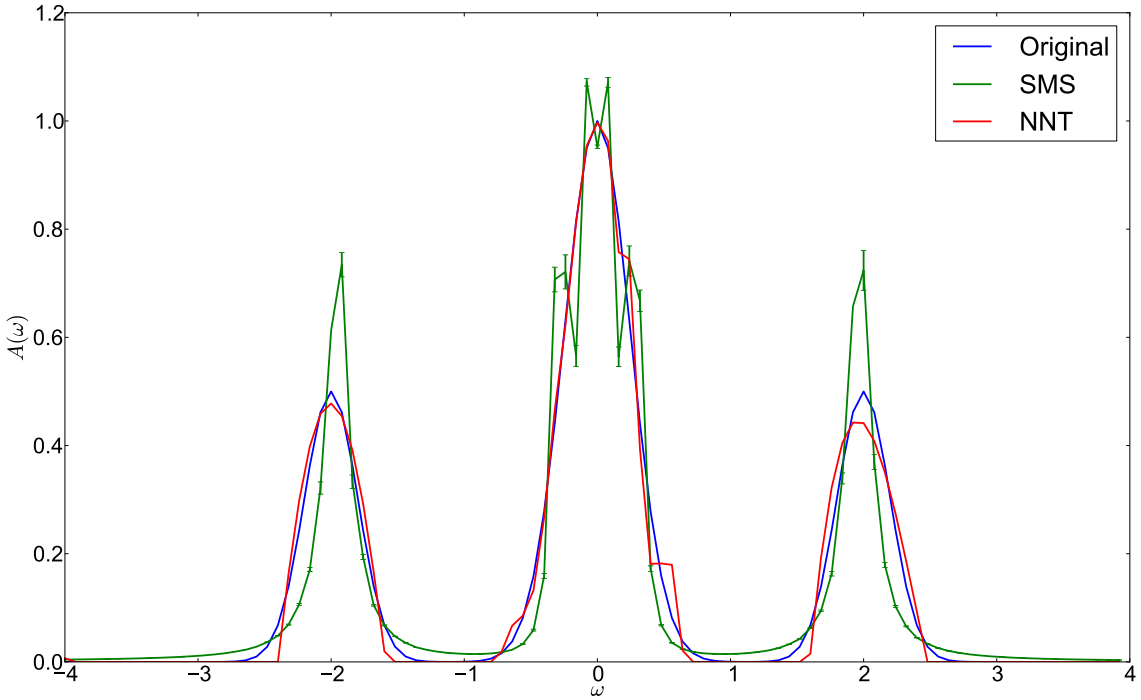

Fig. 2.17.: Stochastic mode sampling (SMS) solution vs. non-negative Tikhonov (NNT) solution with $\alpha = 10^{-4}$ (best parameter according to L-curve method)

CHAPTER 3

SMS Case Study: Optical Conductivity

In this chapter, we apply the SMS method to the analytic continuation of the bosonic optical conductivity. The optical conductivity $\sigma(\omega)$ is related to the Fourier transform of the current-current correlation function $\Pi(\nu)$ by the Fredholm integral equation of first kind

$$\Pi(\nu) = \frac{1}{\pi}\int_{-\infty}^{+\infty} \frac{\omega^2}{\nu^2+\omega^2}\sigma(\omega)d\omega = \frac{2}{\pi}\int_{0}^{+\infty} \frac{\omega^2}{\nu^2+\omega^2}\sigma(\omega)d\omega \tag{3.1}$$

where $\sigma(\omega)$ is always a non-negative and symmetric function, $\Pi(\nu)$ is defined at the bosonic Matsubara frequencies $\nu_n = 2\pi nT$ and T is the temperature. The last equality holds because the integrand is symmetric around zero.

We choose this particular case because Ref. [16] addresses this problem in detail and compares the results of different methods, which provides us with a good starting point to evaluate the quality of our method in comparison to others.

3.1. Test Cases

Ref. [16] uses the following function as an optical conductivity for its tests

$$\sigma(\omega) = \left\{ \frac{W_1}{1+(\omega/\Gamma_1)^2} + \frac{W_2}{1+[(\omega-\epsilon)/\Gamma_2]^2} + \frac{W_2}{1+[(\omega+\epsilon)/\Gamma_2]^2} \right\} \frac{1}{1+(\omega/\Gamma_3)^6} \,. \tag{3.2}$$

Using Eq. (3.1), data $\Pi(\nu)$ is generated. Some noise is then added to the data and the model $\sigma(\omega)$ is reconstructed with different methods using the noisy data.

Ref. [16] lists four test cases. All of them share the following:

- Model parameters: $\Gamma_1 = 0.3$ or $0.6, \Gamma_2 = 1.2, \Gamma_3 = 4, \epsilon = 3, W_1 = 0.3, W_2 = 0.2$
- Bosonic case: Matsubara frequencies are $\nu_j = j\ \nu_1$, where $j = 0, 1, 2, ...$ and ν_1 is related to the temperature by $\nu_1 = 2\pi T$. Temperature is set to $T = 1/15$.
- Data points: data $\Pi(\nu_i)$ is generated for the 60 smallest non-negative frequencies.
- Data noise: relative noise, i.e. noisy data is related to the exact one by $\tilde{\Pi}(\nu_j) = \Pi(\nu_j) * (1 + r_j)$ where r_j is a normal random variable with zero mean and variance σ_0^2. This means that the covaraince matrix of the SMS method is a diagonal matrix whose diagonal element $\mathbf{C}_{j,j}$ is $[\tilde{\Pi}(\nu_j)\sigma_0]^2$.

The test cases differ in the model parameter Γ_1 and the noise variance σ_0^2:

- *Test case 1*: $\Gamma_1 = 0.6, \sigma_0 = 0.01$.
- *Test case 2*: $\Gamma_1 = 0.3, \sigma_0 = 0.01$.
- *Test case 3*: $\Gamma_1 = 0.6, \sigma_0 = 0.001$.
- *Test case 4*: $\Gamma_1 = 0.3, \sigma_0 = 0.001$.

First, we reproduce those test cases and solve them with the SMS method using some initial discretization parameters. Then we consider the effect of different factors on the quality of the reconstruction.

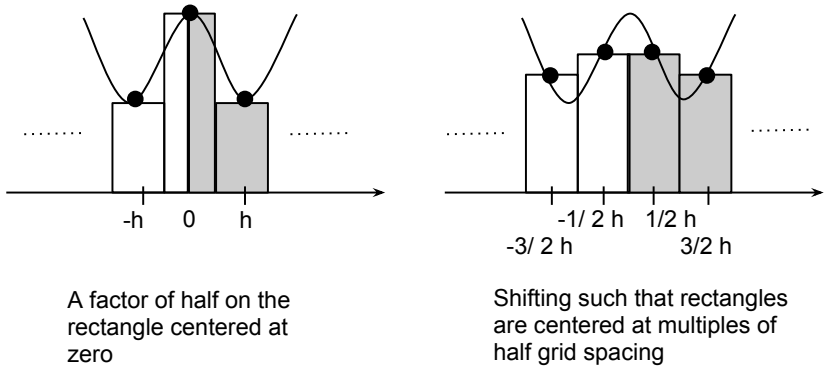

Fig. 3.1.: When we compute the integral of an even function, it is sufficient to consider the positive part only and multiply the result by two. However, we must take care that the symmetry is respected by the numerical quadrature approximating the positive part. For the rectangle rule, there are two options; Either introduce a factor 1/2 on the first rectangle or shift the rectangles such that the left edge of the first rectangle is at zero.

3.2. Preliminary Results

One thing that is not discussed in Ref. [16] is how to discretize the integral on the right-hand side of Eq. (3.1). As a starting point, we take an ω grid in the range $[0, 5]$ with 50 points and use the rectangle rule. To account for the symmetry in the rectangle rule, we should either introduce a factor 1/2 for the rectangle centered at zero or shift the grid by half of the grid spacing (see Fig. 3.1). Both choices give comparable approximations to the integral and none in favorable in this respect. However, we found that the first choice leads to an undesirable effect when solving the inverse problem; the value of the reconstructed model at zero is twice the value it should be (see Fig. 3.2). Our explanation is that the half factor on the first rectangle breaks the smoothness of the matrix representing the kernel. Indeed, by plotting the leading modes of the corresponding matrix, we see that they are smooth except for the first point (see Fig. 3.3).

The SMS results are presented in Fig. 3.4. We have used eight different noisy data samples and shown the SMS solution from each data sample (dashed green). The error bars (which are indeed very small) on each green curve are computed as described in App. C. The blue curve is the average of the green ones. Its error bars are the standard deviation estimated from the eight independent curves. The results are in a surprisingly good agreement with the original model, and they are of the same quality of the best results reported in Ref. [16].

3.3. Noise Effect

We notice from Fig. 3.4 that the SMS method is not resilient to noise i.e. we get different solutions for different noisy samples. There are two notes here regarding the effect of noise.

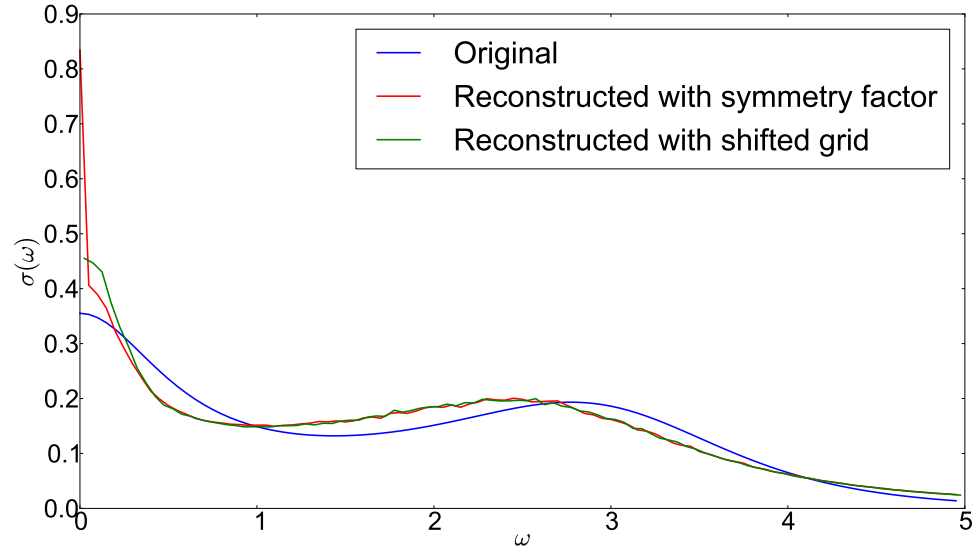

Fig. 3.2.: SMS solution for test cast 1 using two grids: a grid that starts at zero and has a symmetry factor (Red), and a grid that is shifted by half grid spacing (Green). Notice that the grid with the symmetry factor fails at $\omega = 0$.

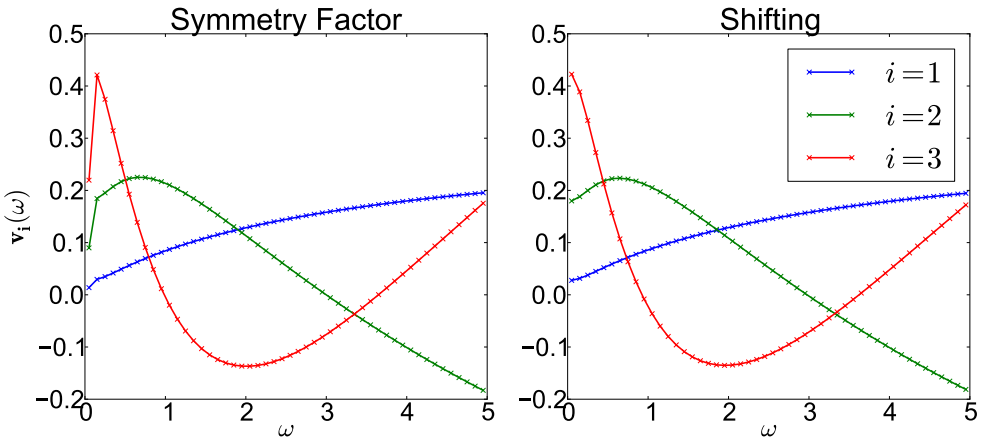

Fig. 3.3.: First three modes (aka right singular vectors) of matrices resulting from discretizing optical conductivity kernel. On the left, the discretization is done by the rectangle rule with a factor 1/2 on the first rectangle. On the right, the discretization is done by rectangle rule with a grid shifted by half the grid spacing. Notice that in the left case, the smoothness is lost at the first gird point which leads to an error in the SMS solution at that point. (see Fig. 3.2)

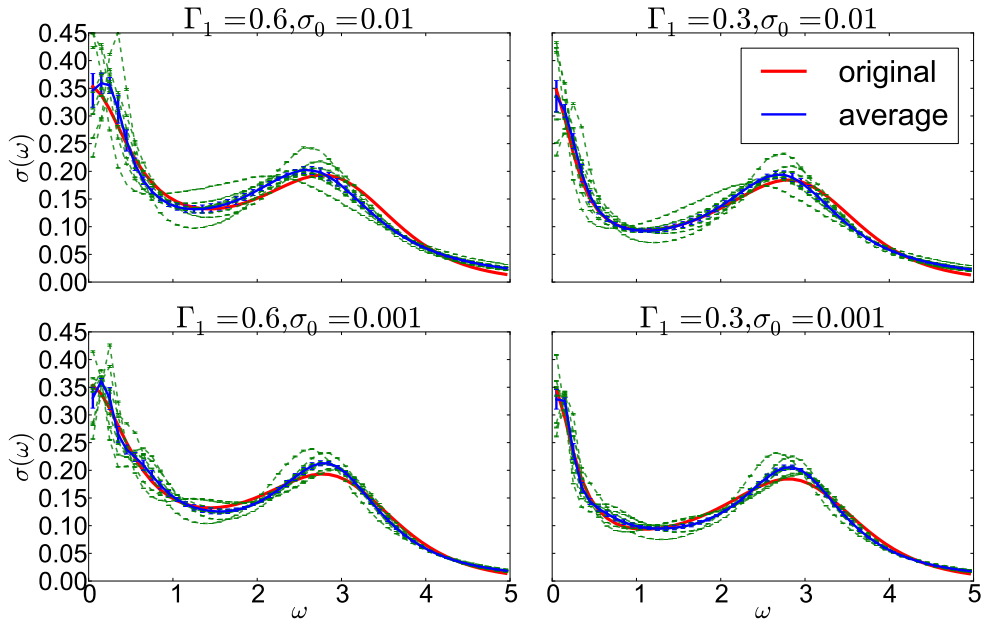

Fig. 3.4.: The results of SMS method applied to the test cases described in Ref. [16]. In each figure, the red line represents the original model while the 8 dashed green lines represent SMS solutions obtained using different noisy data samples. The blue curve is the average of the green ones. The grid is uniform with spacing 0.1 and cutoff 5. Each SMS solution is obtained by a single run of length $2^{23} \approx 8$ million samples.

First, notice that the difference between the reconstructed models is smaller for $\sigma_0 = 0.001$ than for $\sigma_0 = 0.01$. So the smaller the variance of the noise is, the less the difference between the solutions. Second, notice that the difference is larger for smaller ω values. The sensitivity of $\sigma(\omega)$ near $\omega = 0$ can be understood directly from the integral equation; $\sigma(0)$ contributes only for $\nu = 0$ and thus the information about $\sigma(0)$ is contained in only one data value $\Pi(0)$, which makes it more sensitive to noise.[1]

Ideally we would like the solution to be independent of the data noise. If we have several noisy data samples, there are two options to reduce the noise effect. We can apply the SMS method to each sample individually and then average the results. We call this *post-averaging*. Alternatively, we can average the data samples and obtain a more accurate sample and then apply the SMS to this sample. We call this *pre-averaging*. Although the data passed to the SMS in pre-averaging is more accurate than an individual sample, we still pass the covariance matrix of an individual sample, i.e. SMS assumes more noise on the data than there actually is, which helps in reducing the noise effect. In Fig. 3.5, we compare the post-averring and the pre-averaging and we find that they are identical. Ref. [17] applies a similar approach to the maximum entry method.[2]

[1] Do not confuse $\sigma(0)$ with σ_0. The first is value of the model at $\omega = 0$, while the second is the standard deviation of data noise.

[2] A popular method used in solving the analytic continuation problem.

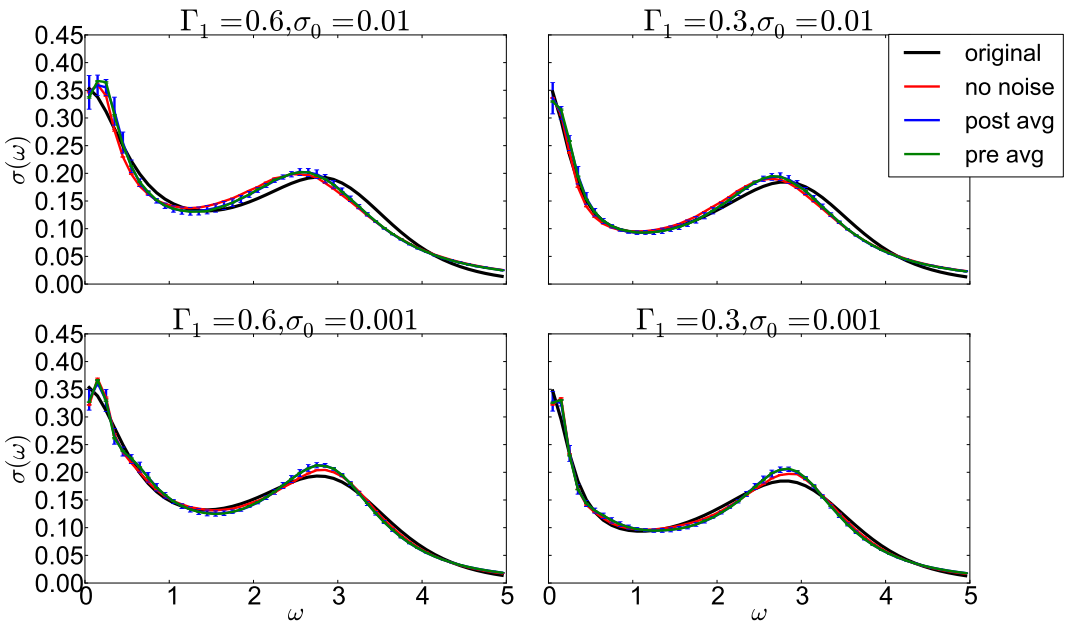

Fig. 3.5.: Comparing the SMS solution using exact data (green) and the average of 8 SMS solutions obtained using 8 noisy data samples (blue). The grid is uniform with spacing 0.1 and cutoff 5.

Fig. 3.5 also shows the solution obtained from the SMS applied to the data without any noise (SMS still uses the usual covariance matrix). This solution matches the other two: pre-averaging solution and post averaging.

Important Note Using this result and in order to reduce the computational work during parametric study, we will use data without noise unless indicated otherwise. This way we include the general behavior resulting from the presence of the noise via the averaging done automatically by the SMS method over a radius of σ_0, but rule out the effect of specific realization of noise.

3.4. Effect of Data Size

Initially we used $\Pi(\nu_i)$ for the 60 smallest non-negative frequencies as in Ref. [16]. To verify that this number of data values (denoted as m) is sufficient, we repeat calculations using $m = 15$, 30, 60 and 120 for test case 3. Fig. 3.6 shows that starting from $m = 30$, the solutions are identical up to error bars. We conclude that for large enough data size, taking more data values does not affect the reconstruction. Therefore, we will keep on using $m = 60$ unless indicated otherwise.

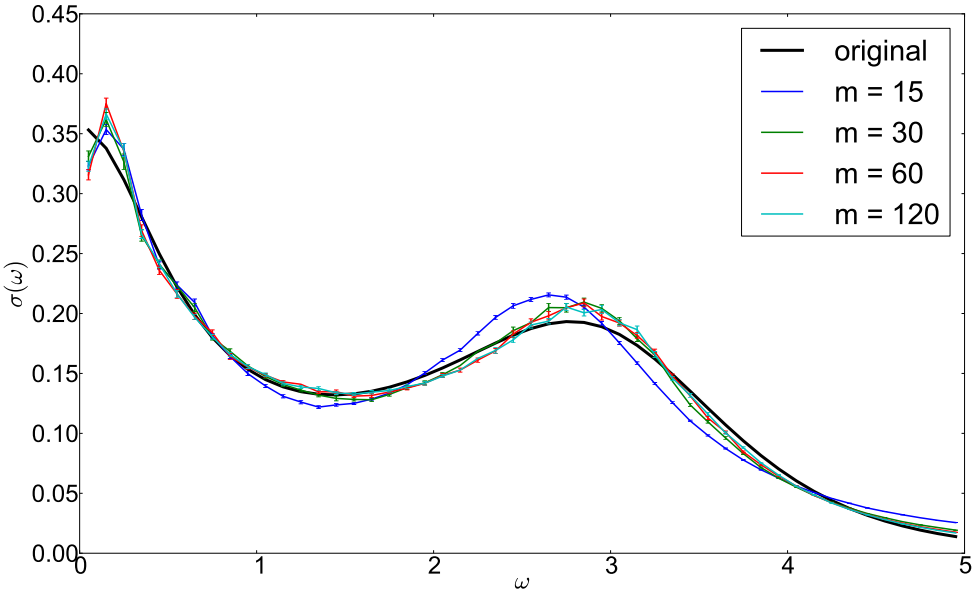

Fig. 3.6.: SMS solutions of test case 1 using different data sizes. Note that starting from $m = 30$, the reconstructed models are identical up to error bars. The grid is uniform with spacing 0.1 and cutoff 5.

3.5. Effect of Systematic Error

In the results discussed above, the data was generated using the same matrix that is used for the SMS method and so we have ignored the effect of the systematic error (discretization and cutoff) on the solution. To study this effect, we need an analytic formula of the data, which is derived next.

Let us define the following integral:

$$I(\nu, a, b, c) = \int_{-\infty}^{+\infty} \frac{\omega^2}{(\nu^2 + \omega^2)[a^2 + (\omega + b)^2](c^6 + \omega^6)} d\omega \ . \tag{3.3}$$

Since the integrand is decaying fast enough, the value of I can be computed by closing the contour of the corresponding complex integral with an infinite semi circle in the upper half plan. This contour integral can then be evaluated by applying the residue theorem:

$$\begin{aligned} I(\nu, a, b, c) = 2\pi i \, [&Res1(i\nu) + Res2(-b + ia) + \\ &Res3(c\ e^{i\pi/6}) + Res3(c\ e^{i\pi/2}) + Res3(c\ e^{i\pi 5/6})\Big] \end{aligned} \tag{3.4}$$

where

$$Res1(\omega) = \frac{\omega^2}{2\omega(a^2 + (\omega + b)^2)(c^6 + \omega^6)} \tag{3.5}$$

$$Res2(\omega) = \frac{\omega^2}{2(\omega + b)(\nu^2 + \omega^2)(c^6 + \omega^6)} \tag{3.6}$$

$$Res3(\omega) = \frac{\omega^2}{6w^5(\nu^2 + \omega^2)(a^2 + (\omega + b)^2)} \ . \tag{3.7}$$

Then the data corresponding to optical conductivity Eq. (3.2) reads

$$G(\nu) = \frac{\Gamma_3^6}{\pi} \left[W_1\Gamma_1^2 \, I(\nu, \Gamma_1, 0, \Gamma 3) + \ W_2\Gamma_2^2 \, I(\nu, \Gamma_2, \epsilon, \Gamma 3) + W_2\Gamma_2^2 \, I(\nu, \Gamma_2, -\epsilon, \Gamma 3) \right] \ , \tag{3.8}$$

which is an algebraic expression that is easily evaluated. Although evaluating this expression involves intermediate complex numbers, the final result is real.

The first source of systematic error is the discretization. As an attempt to reduce the discretization error, we have tried to use a numerical quadrature of an order higher than the rectangle rule; Simpson's rule. One would expect the discretization error of the rectangle rule to be quadratic with grid spacing $\mathcal{O}(h^2)$, while the Simpson's rule to be quartic $\mathcal{O}(h^4)$. To our surprise, this behavior was not observed and moreover the rectangle rule outperformed the Simpson's rule! This turned out to be a special property of the integrand. The discretization error of the rectangle rule [3] has been proven to decreases exponentially with the reciprocal of the grid spacing for integrals over real axis of functions analytic in an open strip containing the real axis (see Refs. [2, 3]). We did observe this exponential decaying behavior for both the rectangle and Simpson's rules and the decay is faster for the rectangle rule (see Fig. 3.7). As a result, we stick with the rectangle rule.

The second source of systematic error is the cutoff on ω. This happens because the integral goes theoretically to infinity, while it is evaluated numerically only up to a maximum ω. Fig. 3.8 shows how the error changes with the cutoff.

Now we can study the effect of the systematic error on the SMS solution. Results are shown in Fig. 3.9. For each of the four cases, we solve using both the analytic data and the numerical data (i.e. data generated by the matrix used for reconstruction) and we do it for three cutoffs: 5, 8 and 11. For cutoff 5, the difference between using analytic data and numerical data is significant, while for cutoffs 8 and 11, the solutions are similar up to error bars. The reason is that the systematic error becomes negligible in comparison with the data noise as the cutoff increases (see Fig. 3.10). It is important to notice that cutoff 5 gives good results only when the numerical data is used, so the preliminary results shown earlier are unreliable. It is also strange that solutions (whether obtained from analytic or numerical data) change with the cutoff; they are actually becoming worse. Since this behavior happens for both types of data, it is not the result of systematic error but rather the grid itself. We study this effect in Sec. 3.7.

[3] or equivalently the trapezoidal rule as the rectangle and trapezoidal rules differ only by half of the function values at the end points which is zero in our case because the function goes to zero at plus and minus infinity.

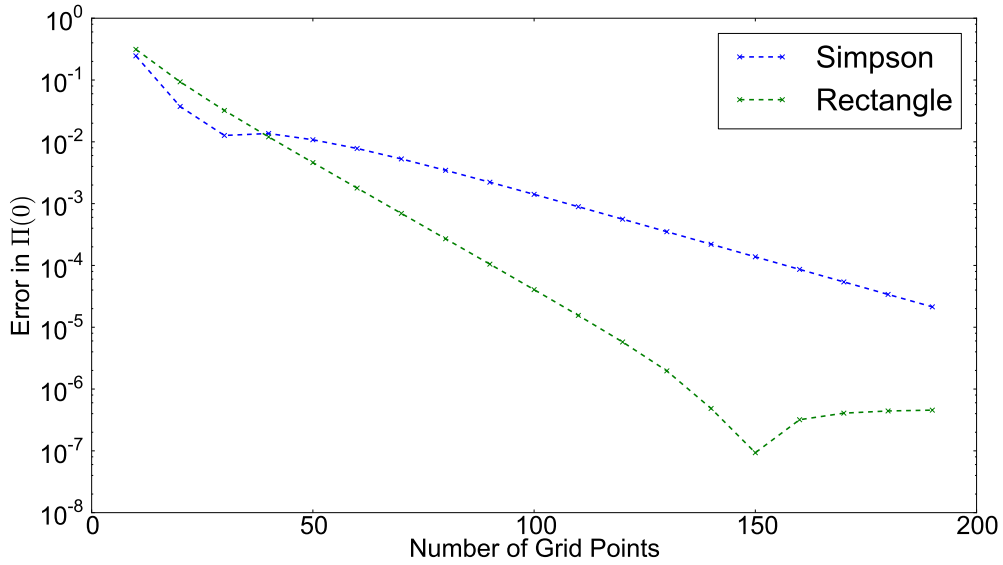

Fig. 3.7.: Relative error in computing $\Pi(0)$ by discretizing Eq. (3.1), plotted against the number of grid points. A uniform grid in the interval $[0, 20]$ is used. Notice that both rules converge exponentially $\mathcal{O}(e^{-\alpha n})$ but the rectangle rule has faster convergence. Also notice that after 160 points, the error reaches a fixed value which is the cutoff error. The model used has $\Gamma_1 = 0.3$.

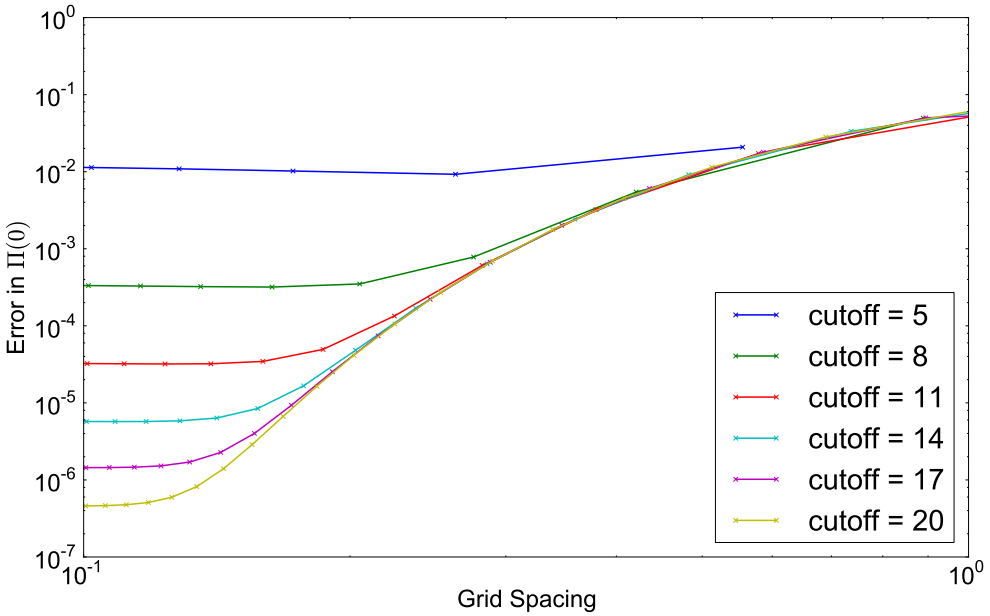

Fig. 3.8.: Relative error in computing $\Pi(0)$ by discretizing Eq. (3.1), plotted against the gird spacing for different cutoff values. A uniform grid in the interval $[0, \text{cutoff}]$ is used. The model used has $\Gamma_1 = 0.3$. Notice that for sufficiently fine grid the error reaches a fixed value which depends on the cutoff. The numbers of grid points required to reach the cutoff error are respectively: 20, 40, 70, 100, 130 and 170.

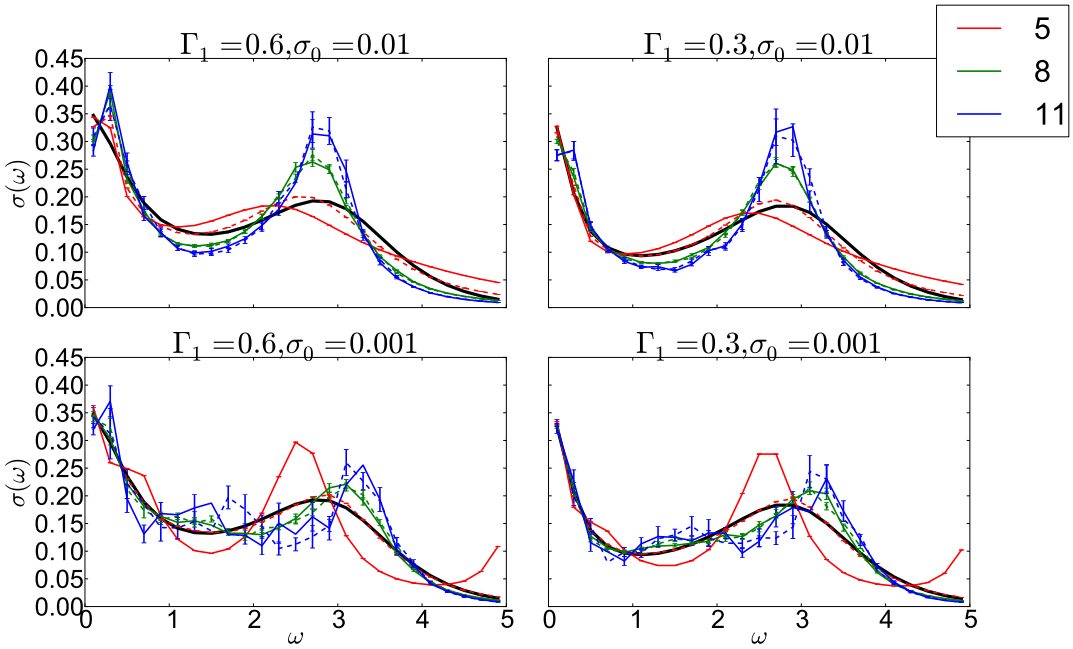

Fig. 3.9.: Thick black line is the exact model. Solid lines are models reconstructed from exact data. Dashed lines are models reconstructed from numerical data. Color represents the grid cutoff: 5 (red), 8 (green) and 11 (blue). For all models, the grid is uniform with spacing 0.2.

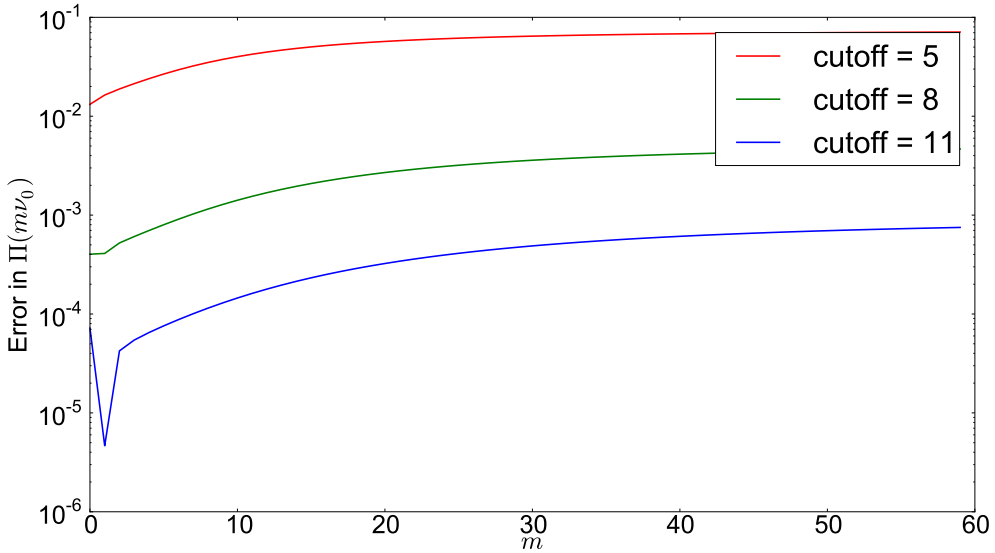

Fig. 3.10.: Relative error in computing $\Pi(\nu)$ for different data values and different cutoffs.

3.6. Kernel Modification

One thing we have noticed when trying different cutoffs is that the larger the cutoff, the larger the correlation time and that for cutoffs greater than $\omega = 15$, the sampling is even "locked" which means that the sampling gets stuck at some sample and no new samples are produced.

To understand the reason behind this, we look at the leading mode[4] as we increase the cutoff. Fig. 3.11 shows that the mode values increase for larger ω and that the modes resulting from different cutoffs are different. This "blowing up" at large ω has drastic effect on the sampling. The leading mode is the only mode without a sign change (due to orthogonality) so its coefficient determines the freedom allowed by the constraint for all other modes. Since the leading mode values increase at large ω while the model values decrease, the coefficient of the leading mode is very small and thus the other modes can only combine in a specific way to satisfy the constraint. This effect gets stronger as we increase the cutoff resulting larger correlation times and eventually locking.

The reason that the leading mode values are increasing with ω is that the kernel values themselves do! Actually, $\lim_{\omega\to\infty} K(\omega,\nu) = 1$. To work around this, we multiply the kernel with some decaying function like $1/(1+\omega^2)$ which brings the kernel values to zero at large ω. Fig. 3.12 shows that the values of the leading mode are also decaying. So instead of solving the original problem with kernel $K(y,x)$

$$\int K(y,x)f(x)dx = g(y)\ , \tag{3.9}$$

we solve the equivalent problem using the modified kernel $K(y,x)m(x)$

$$\int [K(y,x)m(x)]\left[\frac{f(x)}{m(x)}\right]dx = g(y)\ . \tag{3.10}$$

The result of applying the SMS method using the modified kernel will be $f(x)/m(x)$ instead of $f(x)$ which can be fixed by multiplying the result with the modification $m(x)$. The two problems are equivalent as long as the modification $m(x)$ is a strictly positive function.

Fig. 3.13 shows how modifying the kernel helps reducing the correlation time for moderate cutoff and Fig. 3.16 shows how it helps taking large cutoffs that are impossible to treat using the original kernel due to locking.

Fig. 3.14 shows, as expected, that the specific form of modification does not affect the solution as long as this modification is a non-negative function. As a result, we stick with the simplest kernel modification we have tried $1/(1+\omega^2)$.

[4]First right singular vector of the matrix $\mathbf{WK}$ where $\mathbf{W}^T\mathbf{W} = \mathbf{C}^{-1}$ and $\mathbf{C}$ is the data covariance matrix which is diagonal for our test case.

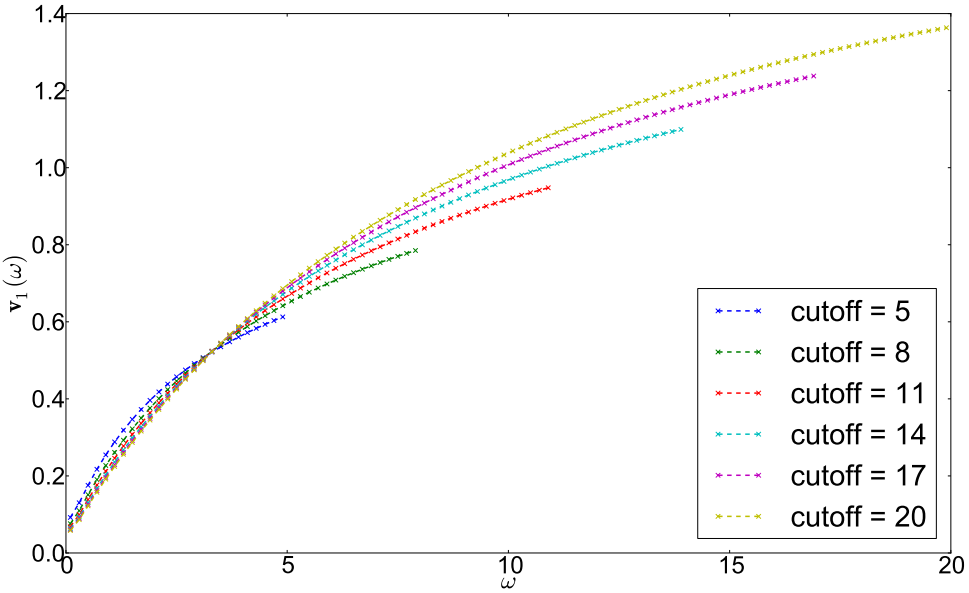

Fig. 3.11.: The leading mode using the original kernel $\omega^2/(\nu^2+\omega^2)$ for different cutoffs. Notice that the values increase for larger ω, and that values resulting from different cutoffs are different.

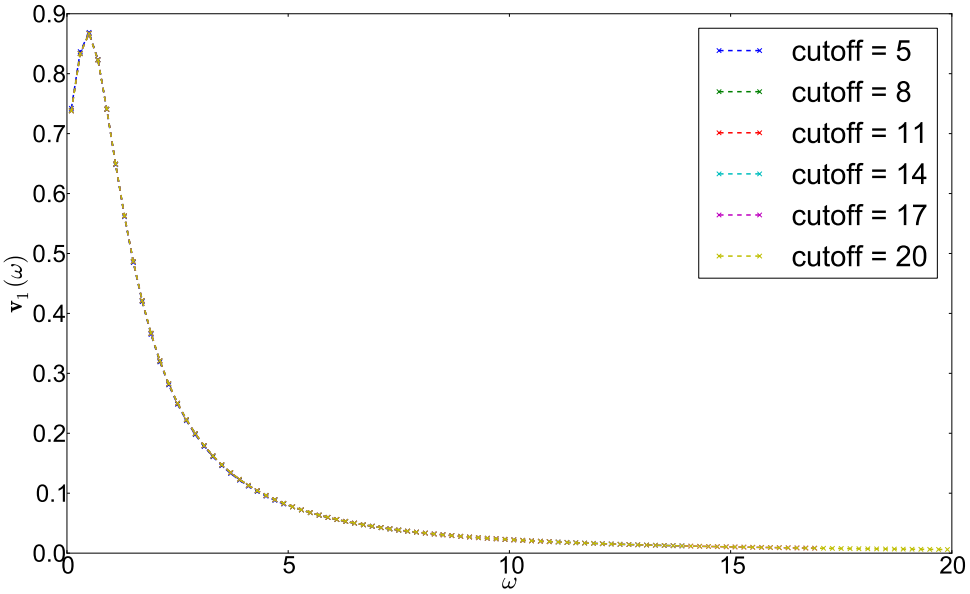

Fig. 3.12.: The leading modes using the modified kernel $\omega^2/[(\nu^2+\omega^2)(1+\omega^2)]$ for different cutoffs.Notice that the values decrease for larger ω and that modes resulting from different cutoffs are identical.

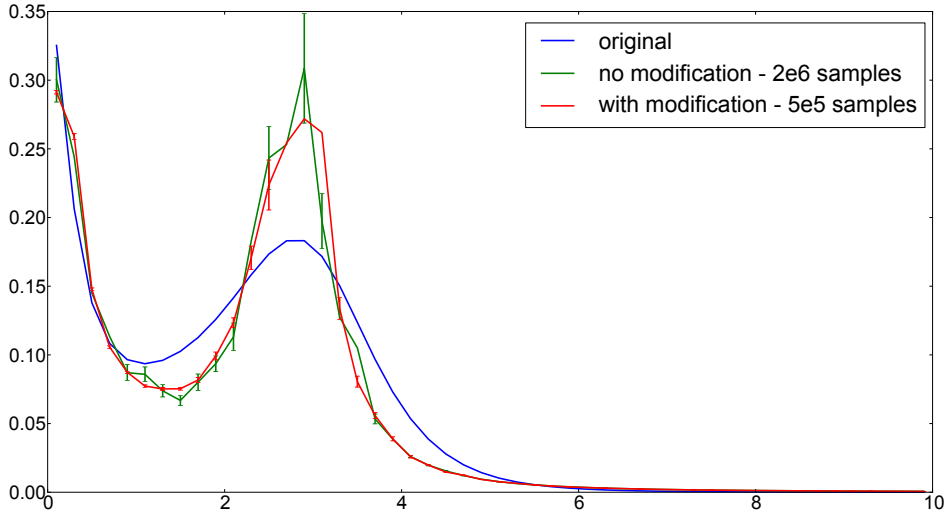

Fig. 3.13.: SMS solutions of test case 2 using the original kernel $\omega^2/(\nu^2+\omega^2)$ (green) and the modified kernel $\omega^2/[(\nu^2+\omega^2)(1+\omega^2)]$ (red). The convergence using the modified kernel was much faster than without modification. Even though the green solution was computed using 4 times the number of samples used for the red one, it still has not reach its quality. This due to the large correlation times using the original kernel compared with the modified one.

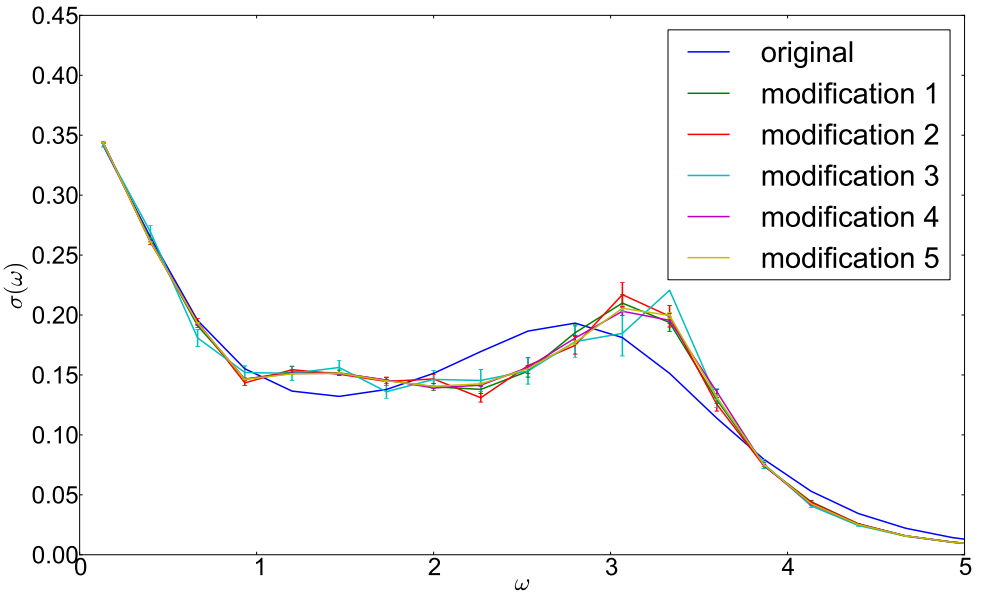

Fig. 3.14.: SMS solutions of test case 1 using different modified kernels of the form $\omega^2/[(\nu^2+\omega^2)(1+(\omega/\omega_0)^p)]$. The parameters of the kernels used to get the shown models are respectively $(p=2,\omega_0=1)$, $(p=2,\omega_0=2)$, $(p=2,\omega_0=4)$, $(p=4,\omega_0=1)$ and $(p=6,\omega_0=1)$. As expected the solutions are the same up to errorbars. Parameters: $\Gamma_1=0.6,\sigma=0.01$. The grid is uniform with spacing 0.2 and cutoff 6.

3.7. Grid Effect

In Sec. 3.5, we verified that when the systematic error is negligible in comparison to the data noise, it has no effect on the SMS solution. However, Fig. 3.10 indicates that the SMS solution (whether obtained from exact or numerical data) dependence on the grid and this depends in not due to the systematic error but rather inherent to the grid itself. First, we present the results using different grids and discuss them. Then we introduce the "Truncated SMS" which does not suffer from grid dependence. Finally, we explain why the SMS solutions depend on the grid and argue what are the appropriate girds to be used.

3.7.1. Uniform Grid

Grid Spacing For a fixed gird cutoff 8, we vary the grid spacing: 0.25, 0.125 and 0.0625. Fig. 3.15) shows that as the grid gets finer the solution becomes smoother without changing its features. This means that SMS solutions converge as the grid becomes finer.

Grid Cutoff For a fixed grid spacing 0.2, we vary the cutoff: 8, 16 and 32. Fig. 3.16 shows that as the grid gets larger, the peaks change and the solution becomes "worse". We also note that increasing the cutoff has effect on the model at small ω.

3.7.2. Nonuniform Grid

The general way of discretizing an integral on a nonuniform grid is changing the integration variable and then taking a uniform gird of the new variable. So we change the variable ω in Eq. (3.1) to a variable z to get

$$\Pi(\nu) = \frac{2}{\pi} \int_{z(\omega=0)}^{z(\omega=+\infty)} \frac{\omega(z)^2}{\nu^2 + \omega(z)^2} \sigma(\omega(z)) \, \frac{d\omega}{dz} \, dz \, . \tag{3.11}$$

Now we discretize this integral uniformly in the variable z and get a nonuniform discretization of the variable ω. Note that the extra factor $\frac{d\omega}{dz}$ is like the weight in Eq. (2.2), so only its square root should be added to the matrix representing the kernel while the other square root is included in model. Since the SVD algorithm gives orthonormal vectors in the Ecludian sense, including only the square root of the weight in the kernel means that the Ecludian norm of the modes is the same as L^2-norm and so we are able to compare modes from different grids with each other. We just have to remember to divide the square root of the weight out of the modes of the solution because it is implicitly included.

Hyperbolic Grid A common non-uniform gird is the logarithmic grid where $\omega(z) = e^{-z}$. However, this transformation does not respect the symmetry of the integrand around zero so we choose instead $\omega(z) = e^{+z} - e^{-z}$ which gives a symmetric integrand

$$\Pi(\nu) = \frac{2}{\pi} \int_0^{+\infty} \frac{\omega^2}{\nu^2 + \omega^2} (e^z + e^{-z}) \sigma(\omega) dz \, . \tag{3.12}$$

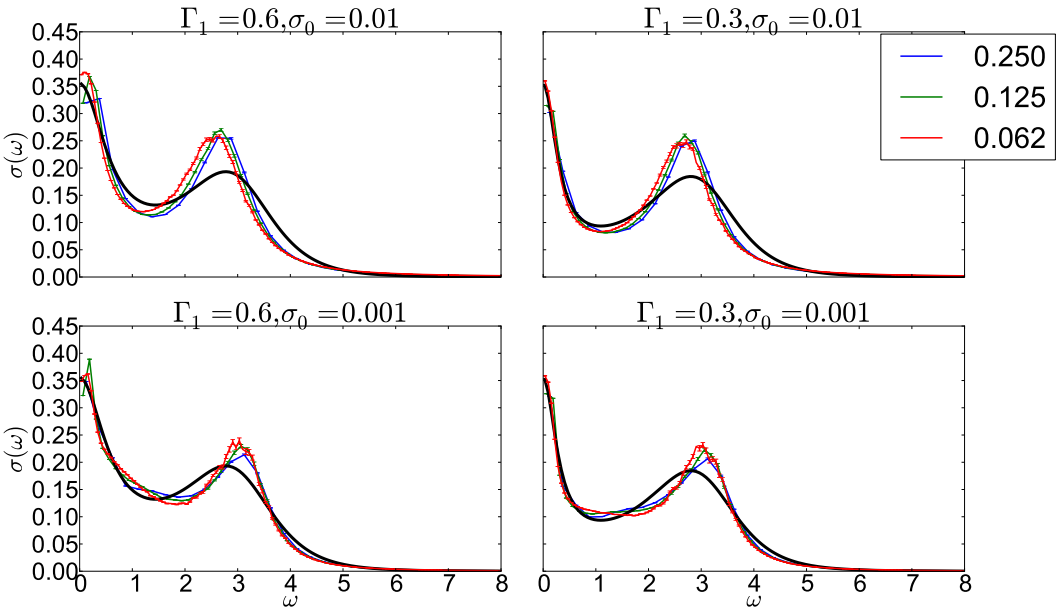

Fig. 3.15.: SMS solutions using uniform grid with different gird spacing: 0.25, 0.125 and 0.0625. Grid cutoff is 8. Each curve is obtained by one SMS run of length $2^{24} \approx 16$ million samples.

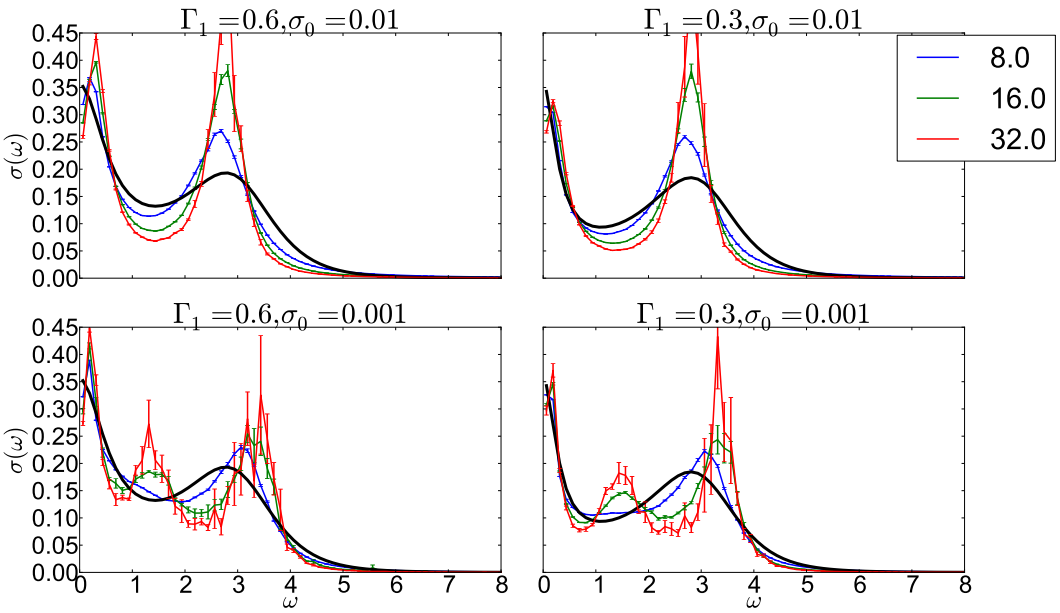

Fig. 3.16.: SMS solutions using uniform grid with different cutoffs: 8, 16 and 32. Grid spacing is $\delta\omega = 0.125$. The result for the first grid is obtained by one SMS run of length $2^{24} \approx 16$ million samples. The result for the second grid is obtained by eight SMS runs, each of length $2^{25} \approx 32$ million samples. The result for the third grid is obtained by eight SMS runs, each of length $2^{24} \approx 16$ million samples.

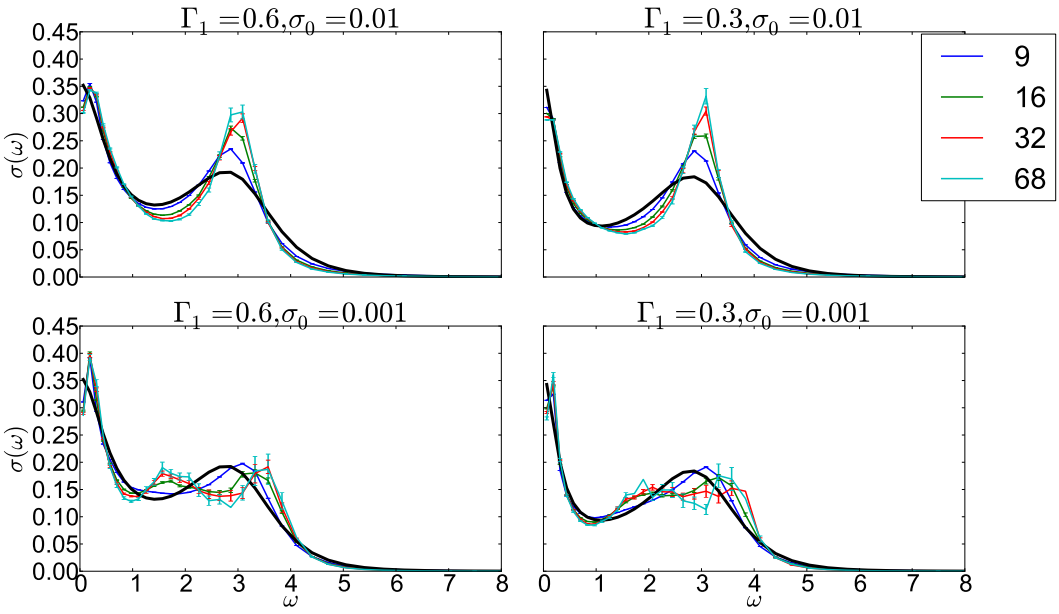

Fig. 3.17.: SMS solutions using hyperbolic grids with different cutoffs: 8.5, 16.0, 32.0 and 67.9. The spacing of all grids is: $\delta z = 0.0625$. Each result for the first two grids is obtained by one SMS run of length $2^{25} \approx 33$ million samples. Each result for the last two grids is obtained by one SMS run of length $2^{26} \approx 67$ million samples.

In Fig. 3.17, we vary the cutoff of this grid. Notice that the dependence on the cutoff is much weaker in comparison to the uniform gird, but it is still present. Besides, the results of cutoffs 16 and 32 are different from that for the uniform gird.

Tangent Grid We try yet another nonuniform gird where $\omega(z) = \tan(z)$ which we call "Tangent Grid". The usage of this grid is motivated by the kernel modification. Using kernel modification $1/(1+\omega^2)$, a point with coordinate ω is given a weight of $1/(1+\omega^2)$. It is reasonable then to distribute the grid points such that points with less weight represent larger areas and vice versa. This can be achieved by requiring the density of grid points to be equal to the kernel modification

$$\frac{dz}{d\omega} = \frac{1}{1+\omega^2} \Rightarrow z(\omega) = \tan^{-1}(\omega) \Rightarrow \omega(z) = \tan(z) \Rightarrow \frac{d\omega}{dz} = \frac{2}{\cos(2z)+1} \tag{3.13}$$

$$\Pi(\nu) = \frac{2}{\pi}\int_0^{\pi/2} \frac{\omega^2}{\nu^2+\omega^2}\left[\frac{2}{\cos(2z)+1}\right]\sigma(\omega)\ \ dz\ . \tag{3.14}$$

Notice that the new variable has a finite interval $z \in [0, \pi/2]$ so there is no cutoff parameter and the cutoff is implicitly specified by the grid spacing or equivalently the number of grid points. Fig. 3.18 shows the relative error in evaluating the integral using the tangent grid. Note that the tangent grid approximates the integral much better than the uniform grid.

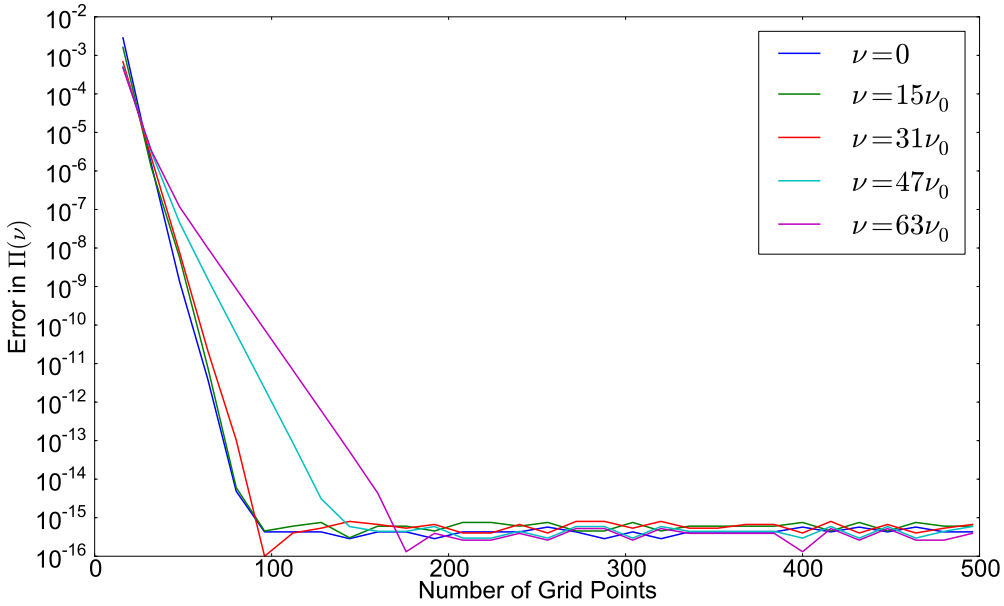

Fig. 3.18.: Relative error in different data values for different resolutions of the tangent grid.

For $\Pi(0)$ for example, we reach numerical accuracy with less than 100 points. Compare this with the uniform gird, where the least error we can get with 100 points is about 10^{-5} for cutoff 14 (see Fig. 3.8). In Fig. 3.19 we show the solutions on a tangent grid using different number of grid points: 32, 64 and 128. The largest ω for these grids are: 40.75, 81.5 and 163, respectively. Except for having higher resolutions, the solutions do not change significantly. This mean that SMS solutions converge for this grid.

3.7.3. Discussion

The previous results show a significant dependence of the SMS solution on the grid cutoff and the grid type. Changing the grid spacing, however, does not change the solution. Clearly, the results of the tangent grid have the best agreement with the original models and they the most reliable, i.e. they are independent of the number of grid points (remember that the cutoff of the tangent grid is determined implicitly by the number of points). In addition, the systematic error reaches numerical accuracy rapidly. However, we do not have a prior justification[5] for using the tangent grid instead of the others. Besides, we need to find an explanation of the strong grid dependence. We will come back to this topic in the next section after introducing the "Truncated SMS".

It is worth mentioning as a side note that the correlation time increases significantly with the cutoff for the uniform grid. This increase also happens for the nonuniform girds, but is

[5] i.e. without knowing the exact data and exact model.

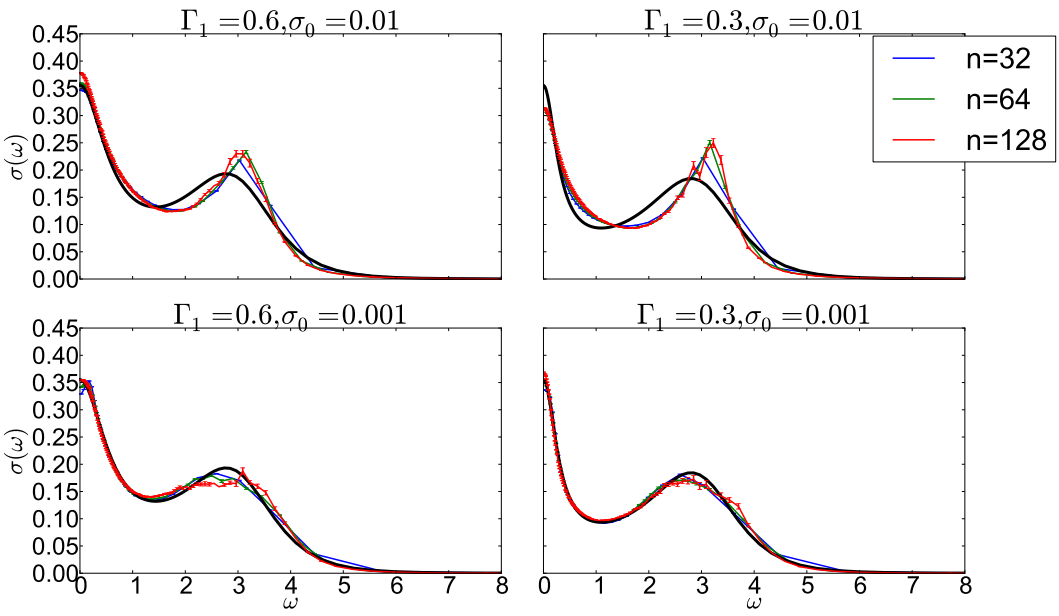

Fig. 3.19.: SMS solutions using tangent grids with different number of points 32, 64 and 128. The largest ω for these grids are: 40.75, 81.5 and 163, respectively.

weaker. The reason for this behavior is that the correlation time increases with the number of active constraints. As the cutoff is increased, more and more grid points are added for large ω where the model is near zero. Maintaining the non-negativity of those points is harder than maintaining it for points of small ω where the model values are away zero. Since nonuniform grids reach larger cutoffs with less number of grid points, this effect is weaker for nonuniform grids than for uniform ones.

3.7.4. Truncating Free Modes

Since the modes are the building blocks of the SMS method and in order to understand the grid effect on the SMS solution, we need to understand the relation between the modes and the grid. The theory of compact operators on Hilbert space tells us that under certain conditions, the integral operator $\int dx\, K(y,x)$ can be expanded using the so-called *Singular Value Expansion*(SVE). We can think of it as a generalization of the SVD concept from matrices to operators. The main difference is that SVE gives continuous singular functions instead of discrete singular vectors. The theory also says that the SVD of the matrix approximating an operator will be an approximation of the SVE of that operator.

This means that whatever grid we choose to discretize the kernel, the singular values and modes of the resulting matrix should converge to the singular values and modes of the kernel as the grid gets larger and denser.[6] However, the SVD algorithm can only compute

[6]For this result to hold, the operator should be compact. We have already seen in Fig. 3.11 that the leading

singular values whose ratio to the largest one is above the machine epsilon while the rest are set zero (or machine accuracy). Moreover, since determining a mode requires determining its singular value, SVD can only finds the modes corresponding to non-zero singular values (non-free modes) while the other modes (free ones) are arbitrary set of orthonormal vectors that make the modes a complete basis. As result, only the non-free modes can converge to the kernel modes and become independent of the grid while the free ones change from one grid to another.

In Fig. 3.20 and Fig. 3.21, we compare the non-free and free ones for three types of girds with similar cutoffs. A uniform grid with cutoff 32 and spacing $\delta\omega = 0.125$ (red curves in Fig. 3.16), a hyperbolic grid with cutoff 32 and spacing $\delta z = 0.0625$ (red curves in Fig. 3.17), and a tangent grid of 32 points which has a maximum ω of 40 (blue curves in Fig. 3.19). Notice that the free modes are not only different for different grids but also have different support. For the uniform grid, for example, the free modes represent oscillations spread uniformly over ω while for the tangent grid, they are compressed near zero.

For this reason it seems reasonable to exclude the free modes from the sampling and simply set their coefficients to zero. Another advantage is that the number of non-free modes r is very small in comparison to the total number of modes n (which equals the number of grid points) and it does not increase as n increases (once n is large enough). This means that sampling only the non-free modes and neglecting the others would reduce the computational time considerably. First, the cost of producing one sample is reduced because we need to sample fewer modes. Second, the correlation time is reduced because the integration space is smaller and thus we need fewer samples. As an example, a uniform grid of 512 points has only 32 non-free modes.

We call sampling using only non-free modes, "Truncated SMS". Fig. 3.22, Fig. 3.23, Fig. 3.24, Fig. 3.25 and Fig. 3.26 show the results of truncated SMS for different grids. We notice that the for large and dense enough girds, the results are independent of the grid because the truncated SMS samples only non-free modes which converge systematically as the grid becomes larger and denser. On the other hand, we see that the truncated SMS leads to extra oscillations that do not exist in the original model or "good quality" SMS solutions (Fig. 3.19 and Fig. 3.15). The reason is that full sampling uses free modes which give the model extra degrees of freedom to satisfy the constraint without affecting the quality of data fitting. While with truncated sampling the models have to satisfy the constraint with the non-free modes only, leading to an overall worse fitting of the data.

3.7.5. Conclusion

Truncated SMS has two advantages over the full SMS. First, it is independent of the grid, and second, it is much faster (especially for dense grids). The disadvantage, however, is that it introduces oscillations that are not present in the original model. Those oscillations can be canceled out using the free modes. However, the free modes depend largely on the

mode of the original kernel $\omega^2/(\nu^2+\omega^2)$ changes as we make the grid larger, and so the corresponding operator is not compact. However, when the kernel is modified such that it decays at large ω, the leading mode (see Fig. 3.12) and all other modes corresponding to nonzero singular values do converge.

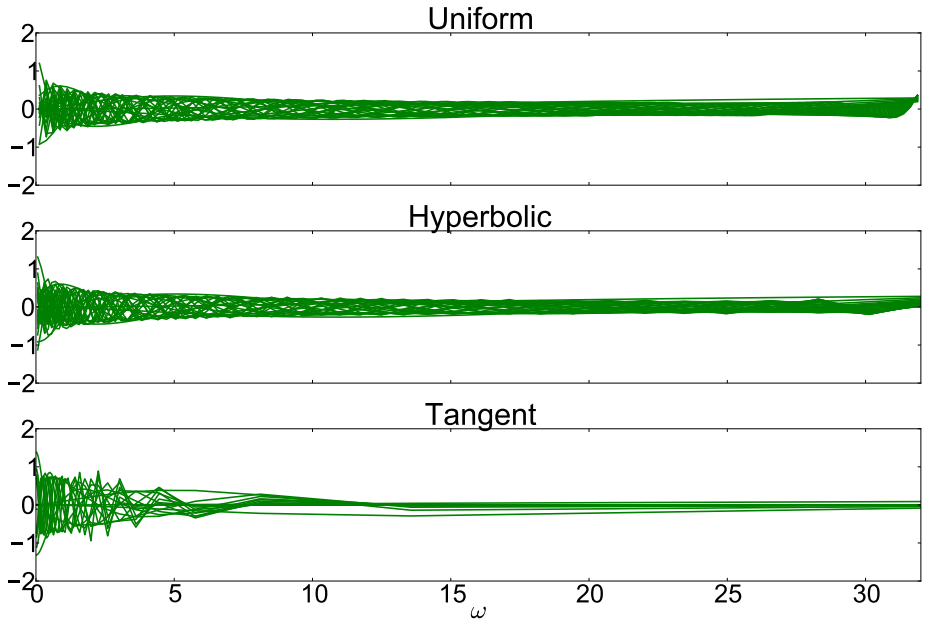

Fig. 3.20.: The non-free modes of different grids: uniform, hyperbolic and tangent. Notice that the non-free modes, unlike the free ones, are similar for different grids. Remember from the description following Eq. (3.11) that we show here the modes obtained by SVD, divided by the square root of the weight.

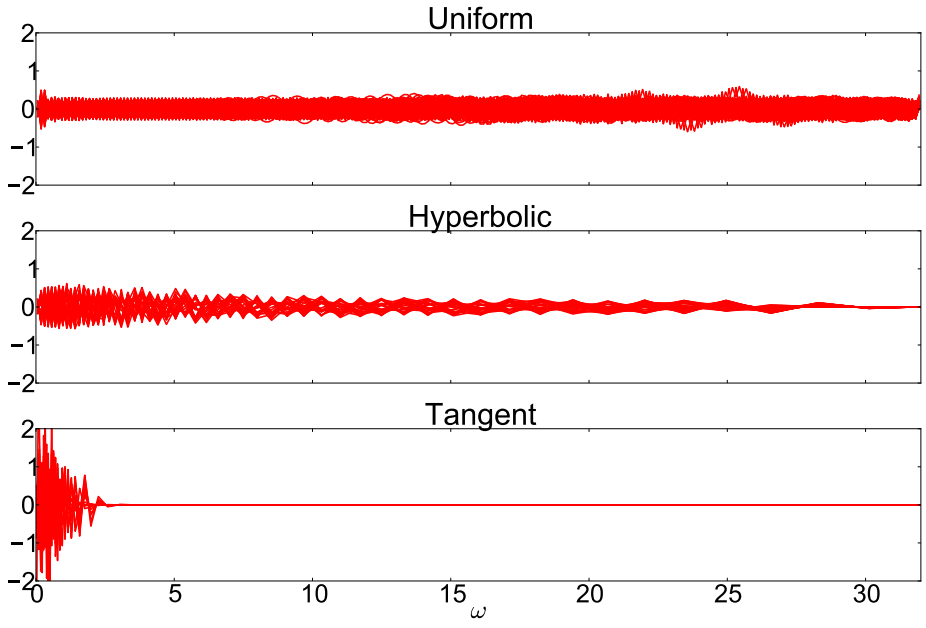

Fig. 3.21.: The free modes of different grids: uniform, hyperbolic and tangent. Notice that the free modes are completely different for different grids. Remember from the description following Eq. (3.11) that we show here the modes obtained by SVD, divided by the square root of the weight.

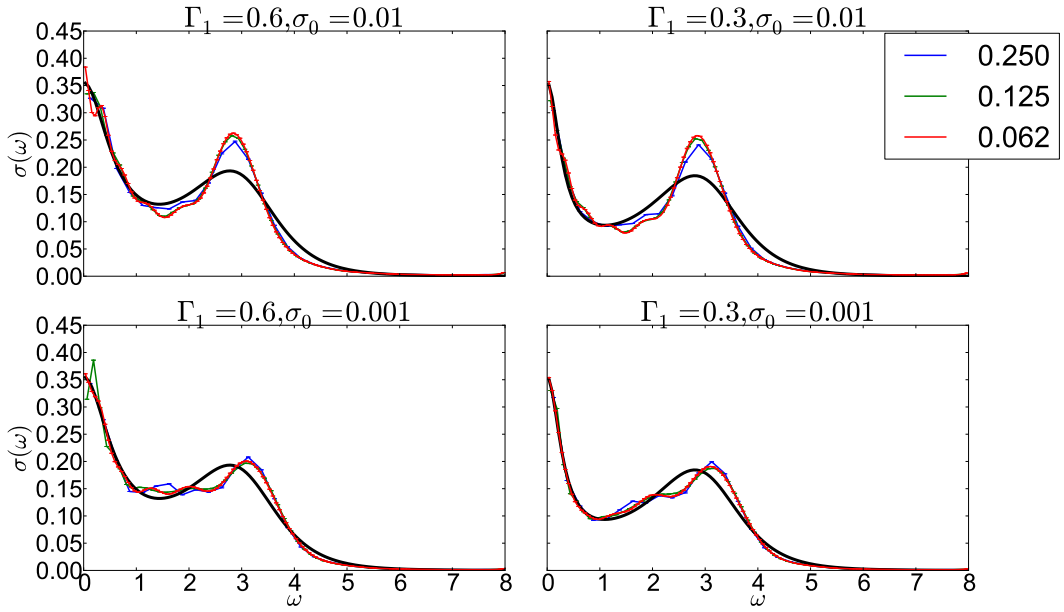

Fig. 3.22.: Truncated SMS solutions using uniform grid with different grid spacing values: 0.25, 0.125 and 0.0625. Grid cutoff is 8. The results of each grid are obtained by one SMS run of length $2^{24} \approx 16$ million samples.

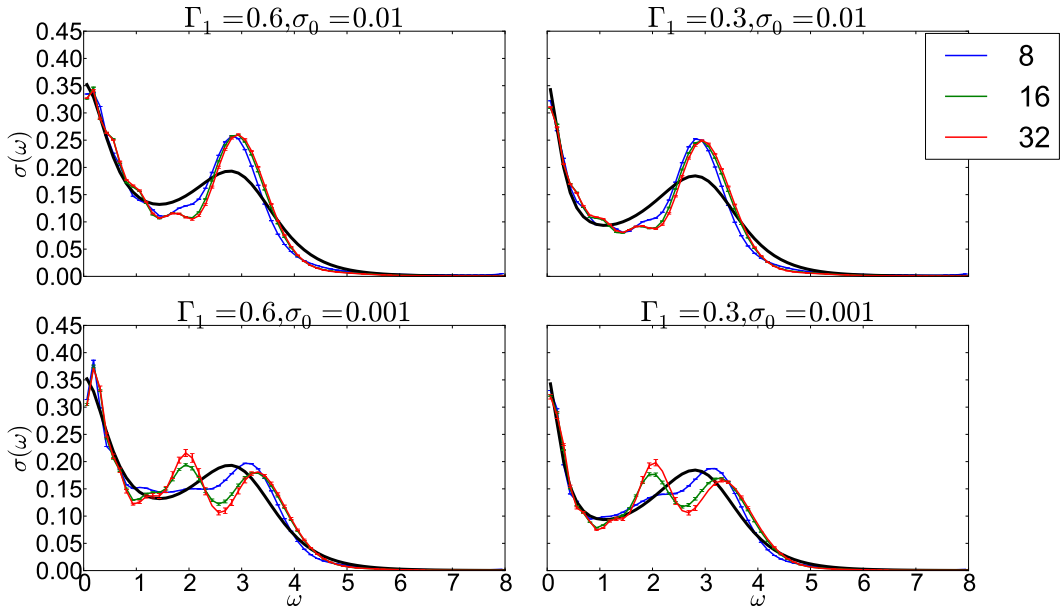

Fig. 3.23.: Truncated SMS solutions using uniform grid with different grid cutoffs: 8, 16 and 32. Grid spacing is $\delta\omega = 0.125$.

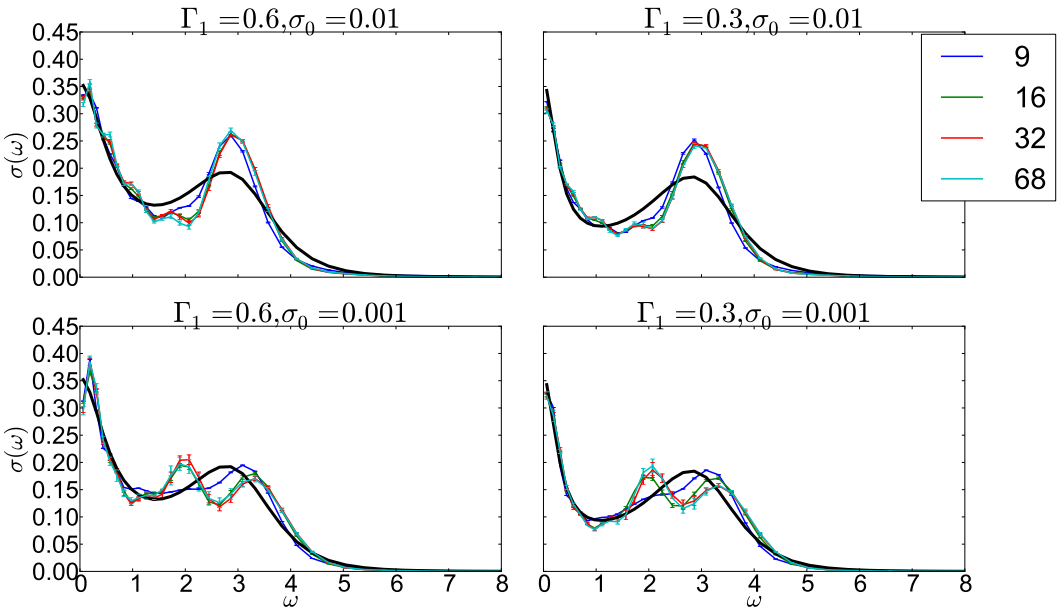

Fig. 3.24.: Truncated SMS solutions using hyperbolic grids with different cutoffs: 8.5, 16.0, 32.0 and 67.9. The spacing of all grids is: $\delta z = 0.0625$.

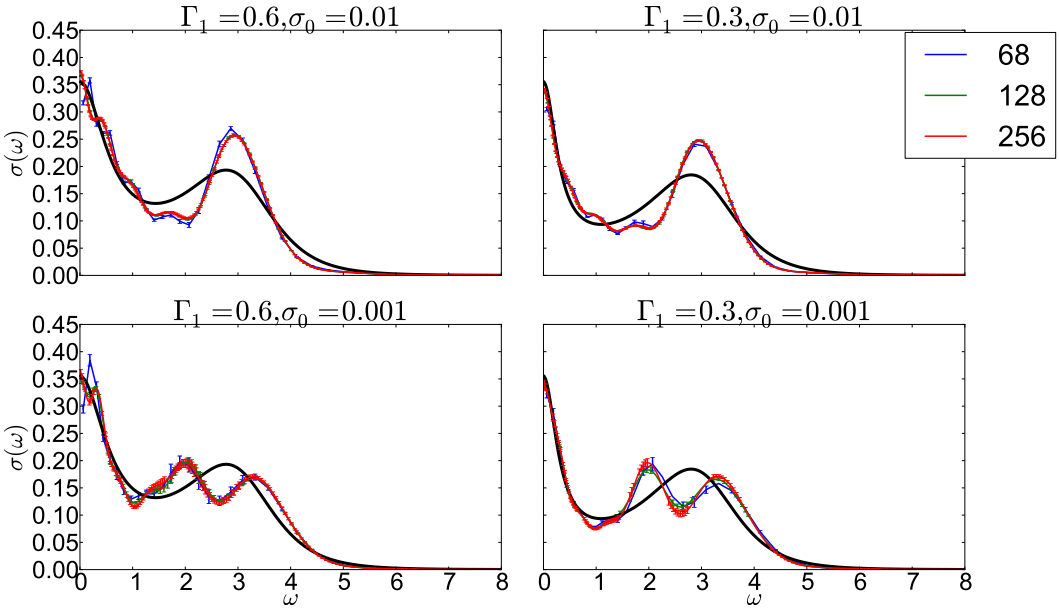

Fig. 3.25.: Truncated SMS solutions using hyperbolic grids with different number of points: 68,128,256. The maximum ω is about 32.

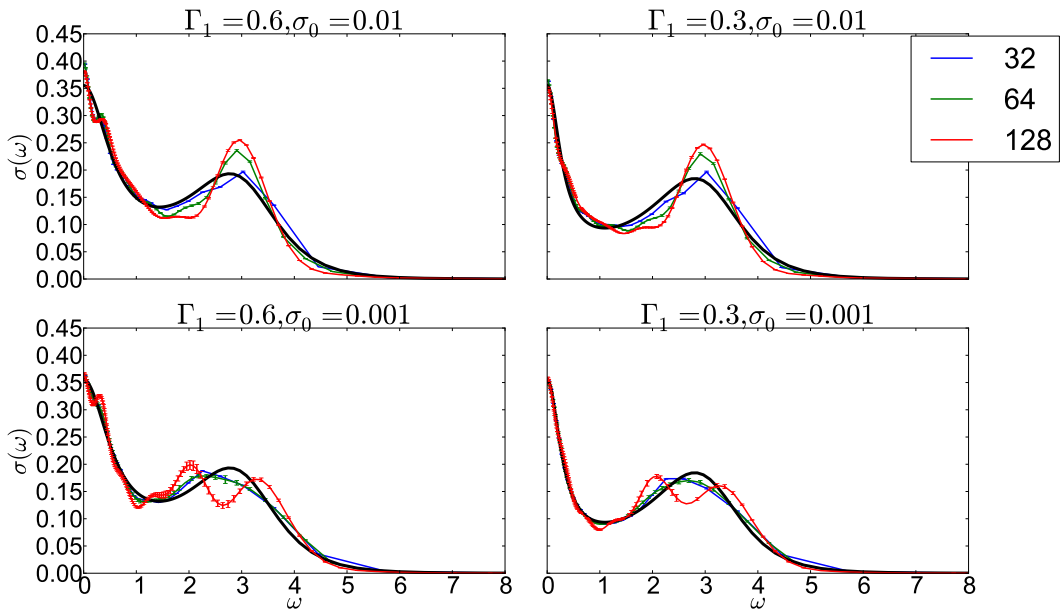

Fig. 3.26.: Truncated SMS solutions using tangent grids with different number of points 32, 64 and 128. The largest ω for these grids are: 40.75, 81.5 and 163, respectively.

grid and we should choose the "right" free modes. The free modes of the uniform grid represent oscillations spread uniformly over the whole grid (see Fig. 3.21). When the grid has small cutoff, the free modes are concentrated in the same region where the model has structure and so they cancel effectively the oscillations of the truncated SMS solution (see blue curve of Fig. 3.16). When the cutoff is increased, the oscillations of free modes are spread equally at small and large ω. Since the model is decaying for large ω, the constraint at large ω allows the free modes smaller permissible intervals. So as we increase the cutoff of the uniform grid, the role of the free modes is diminishing and the solutions become worse. This explains why we get two peaks in the lower plots of Fig. 3.16 as we increase the cutoff [7] (compare Fig. 3.16 with Fig. 3.23). With the hyperbolic grid, this effect is weaker (see Fig. 3.21), because the free modes have less oscillations for large ω than for smaller ω. The effect is almost non-existing for the tangent gird, because the domain of the free modes is concentrated at small ω and extends very slowly as we increase the number of grid points. In this sense, the tangent grid is the best grid (among the ones we tried) for performing the calculations of our test cases. The only problem is that there are other "good" grids. For example we can set the density of points to an exponentially decaying function $dz/d\omega = 1/(1+\omega^p)$ or any other function that concentrate points at small ω, and then follow the same steps we have used to derive the tangent grid. All these choices lead to free modes decaying for large ω but they are systematically different and we expect them to lead to different SMS solutions. The question of which non-uniform grid is better remains open.

[7] However, the reason for getting sharper peaks in the upper plots of Fig. 3.16 is still unknown.

Since in practice, the solution is not known beforehand, we suggest using truncated SMS method on a uniform grid to detect the region where the model has features. This may require trying larger and denser grids till the results are not changed. Once we know the region of interest, we choose a non-uniform grid that concentrate points there and we run the full SMS method on that grid. One may be satisfied with the truncated SMS solution which is grid independent, but as we have shown, it would contain superficial features and oscillations that full SMS helps removing "if a proper grid is used".

3.8. Comparison With Other Methods

Ref. [16] provides results using the following methods: Pade Approximation, Truncated SVD, Stochastic Sampling and Maximum Entropy. Unfortunately, Ref. [16] does not specify the grid used or whether the data used is analytic or numerical. Since we have seen that those are major factors, comparing our results with theirs should be done with caution. If they had used the numerical data with small cutoff, then we should compare their results to Fig. 3.4. The other extreme is analytic data with very large cutoff, then the results should be compared to Fig. 3.19.

Instead of repeating their work, we compare SMS results with other methods: non-negative Tikhonov and noisy kernel method (see Ref.[14] for information on the later method). Since results may depend on the grid, we compare for two grids: a uniform space grid with cutoff 8 and spacing 0.25, and a tangent dense grid with 128 points which extends till $\omega = 160$. The results for the uniform grid are shown in Fig. 3.27. Notice how the noisy kernel method performs very well. Also notice that for non-negative Tikhonov, the values are clamped to zero instead of approaching it smoothly which is a typical feature of this method. Fig. 3.28 shows the results for the tangent grid. Here the SMS method clearly outperforms the other two especially for small ω.

3.9. Final Results

As we have argued before, the best grid is the one which concentrates points where the model is concentrated. The tangent gird already does a good job (see Fig. 3.19) but we would like to generalize it. The tangent gird resulted from requiring the density of grid points to be $1/(1+\omega^2)$ which decays rapidly for $\omega > 2$. By a simple modification we can parameterize it with ω_0 such that it decays rapidly for $\omega > 2\omega_0$.

$$\frac{dz}{d\omega} = \frac{1}{1+(\frac{\omega}{\omega_0})^2} \Rightarrow z(\omega) = \omega_0 \tan^{-1}(\frac{\omega}{\omega_0}) \Rightarrow \omega(z) = \omega_0 \tan(\frac{z}{\omega_0}) \Rightarrow \frac{d\omega}{dz} = \frac{2}{\cos(\frac{2z}{\omega_0})+1} \tag{3.15}$$

$$\Pi(\nu) = \frac{2}{\pi}\int_0^{\omega_0\pi/2} \frac{\omega^2}{\nu^2+\omega^2}\left[\frac{2}{\cos(\frac{2z}{\omega_0})+1}\right]\sigma(\omega)dz\ . \tag{3.16}$$

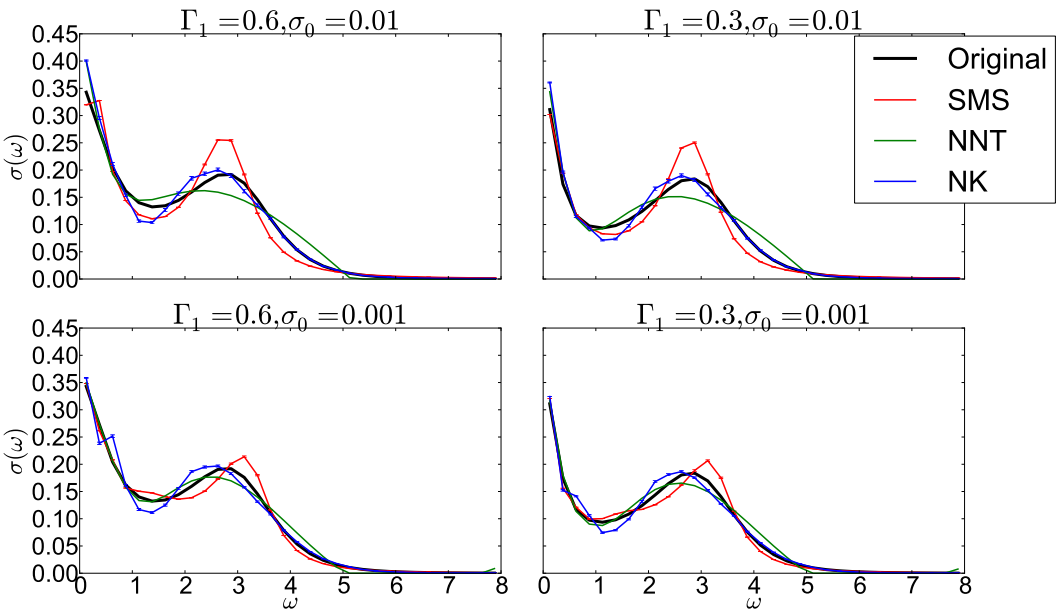

Fig. 3.27.: Comparing SMS method (red), constraint Tikhonov (green) and noisy kernel method (blue) for a sparse uniform grid. Grid cutoff is 8. Grid spacing is 0.25. Analytic data is used and all methods are provided with the same noisy samples.

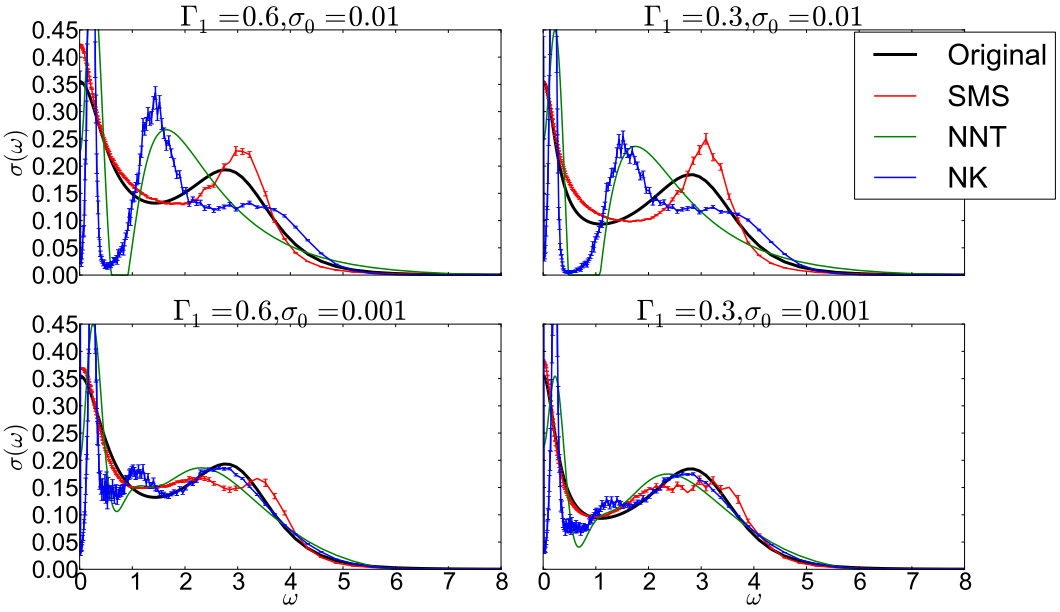

Fig. 3.28.: Comparing SMS method (red), non-negative Tikhonov (green) and noisy kernel method (blue) for a dense tangent grid. Number of grid points is 128. Analytic data is used and all methods are provided with the same noisy samples.

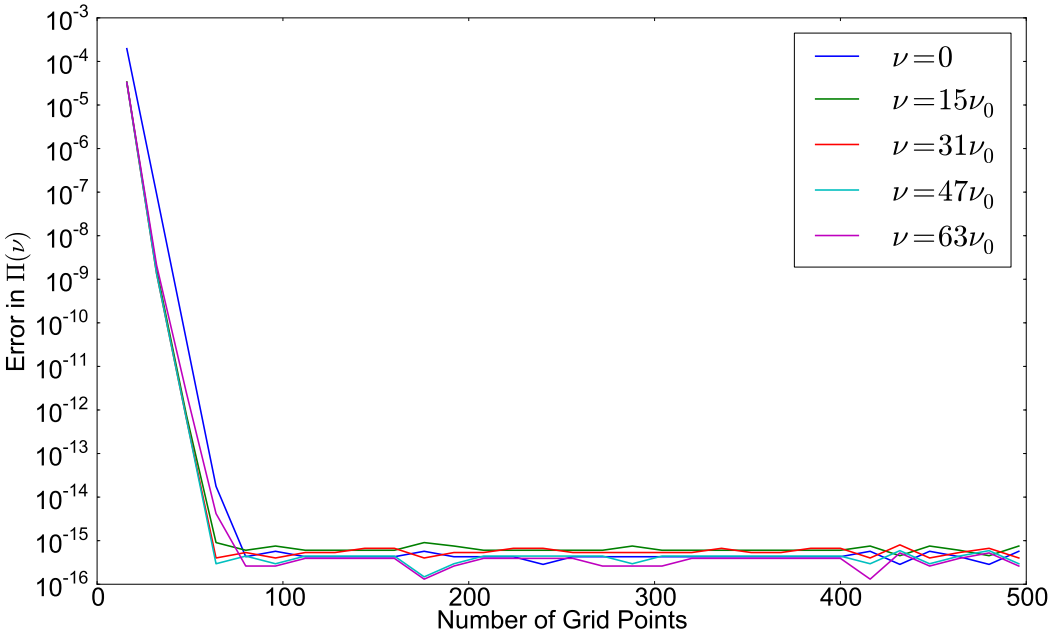

Fig. 3.29.: Relative error in different data values for different resolutions of the tangent grid with $\omega_0 = 2.5$. Compare it with Fig. 3.8 where $\omega_0 = 1$.

Since our model is concentrated till about $\omega = 5$, it is better to choose $\omega_0 = 2.5$ rather than $\omega_0 = 1$. Fig. 3.29 shows the data error using this new tangent grid. Notice how the error goes to zero faster than for the old tangent grid which had implicitly $\omega_0 = 1$ (see Fig. 3.18).

In Fig. 3.30 we show the SMS solutions using what we consider the "optimal" grid; tangent grid with $\omega_0 = 2.5$. We show the results for 64 gird points (blue) and 128 gird points (green). The solutions appear converged with respect to the number of grid points except near $\omega = 0$ where more resolution is needed. Fig. 3.31 shows the solutions using 8 noisy samples (green) and their average (blue) on the grid of 64 points. Comparing the average with the blue solution in Fig. 3.30 (which is computed using exact data) confirms that averaging solutions from different noisy samples is equivalent to solving with exact data.

Although we studied the effect of data size and kernel modification earlier for the uniform grid and found no effect, we repeat this for double checking using the previous grid of 128 point. Fig. 3.33 shows that the results are converged with respect to data size. Fig. 3.32 shows that the results are independent of kernel modification.

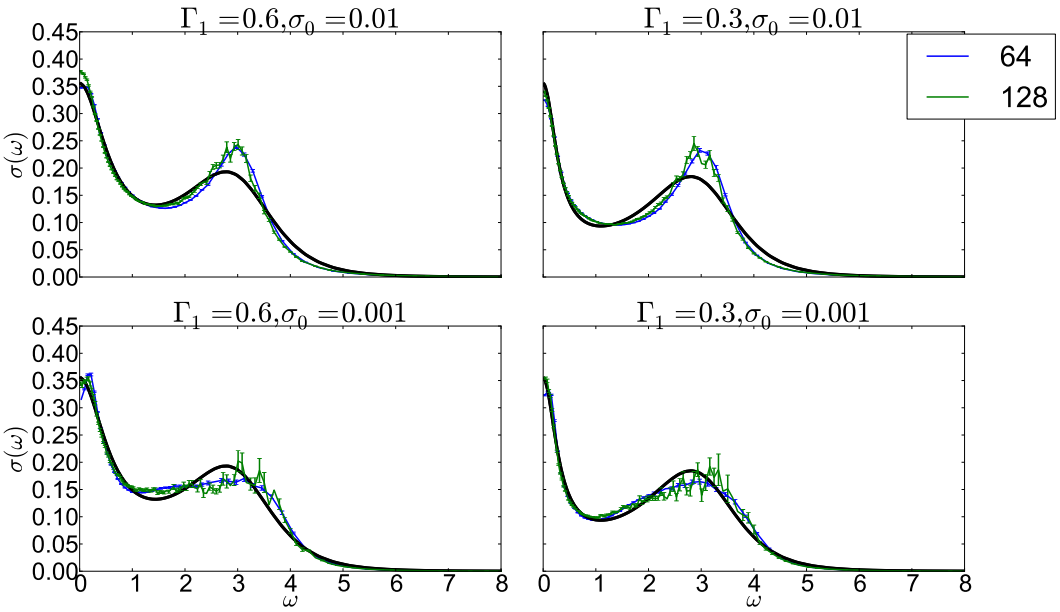

Fig. 3.30.: SMS solution on what we consider the optimal gird; a tangent grid with $\omega_0 = 2.5$ and 64 points (blue) and 128 points (green). The largest ω of those grids is about 200 and 400 receptively.

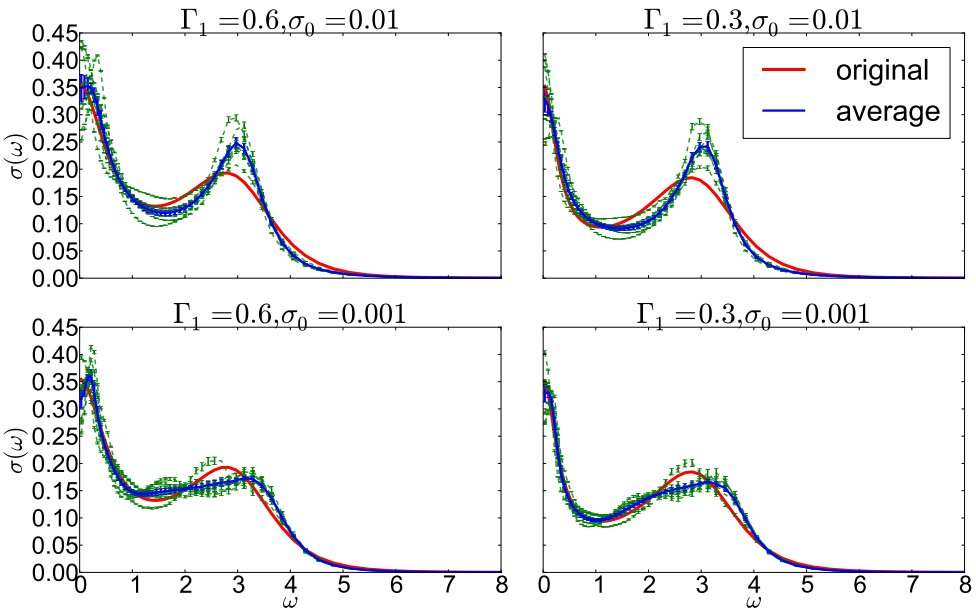

Fig. 3.31.: In each figure, the red line represents the original model while the 8 dashed green lines represent SMS solutions obtained using different noisy data samples. The blue curve is the average of the green ones. The grid is tangent with 64 points and $\omega_0 = 2.5$. Each SMS solution is obtained by a single run of length $2^{25} \approx 33$ million samples.

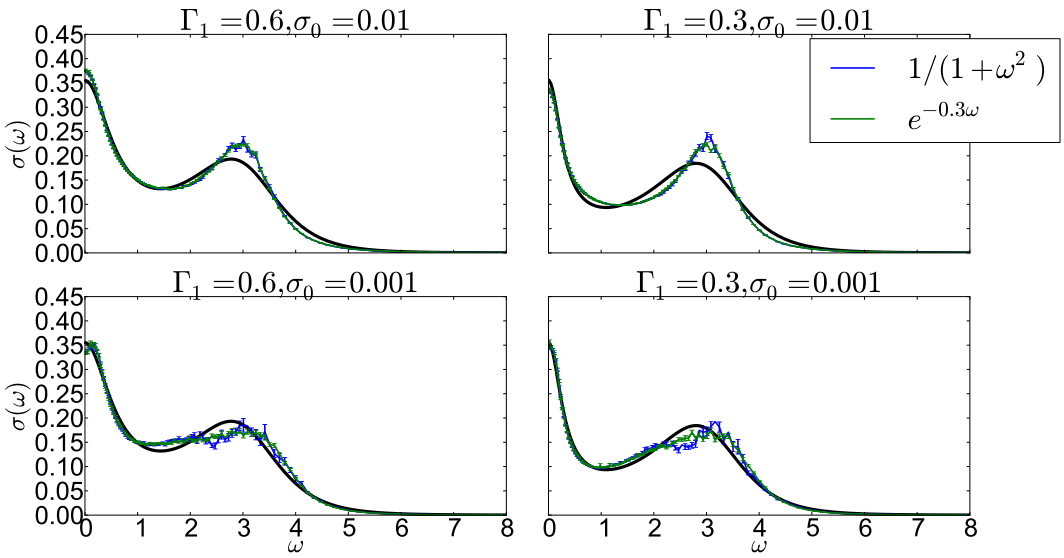

Fig. 3.32.: Comparing the results using different kernel modifications: $1/(1+\omega^2)$ (blue) and $e^{-0.3\omega}$ (green) on the tangent grid with $\omega_0 = 2.5$ and 128 grid points. As expected, the results are independent of the modification.

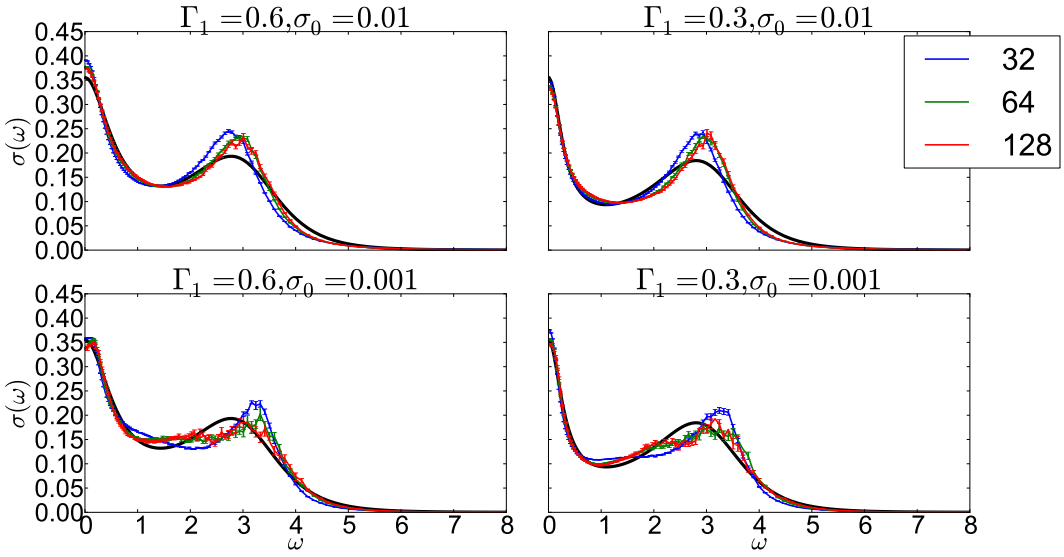

Fig. 3.33.: SMS solutions for different data sizes m. The gird is a tangent grid with $\omega_0 = 2.5$ and 128 points. The results looks converged with respect to data size.

3.10. Future Work

As we have seen, the choice of the proper non-uniform grid is not unique and the SMS results are slightly different between different grids. We believe that the results of different grids can converge to the same results by choosing only a specific subspace of the one spanned by the free modes. We argued previously that only the non-free modes converge to the kernel modes while the free ones depend totally on the grid. But this distinction is based on the machine accuracy which is an arbitrary value and if we were able to increase the machine accuracy, we would obtain more of the non-free modes without changing anything else. This means that for large and dense enough grids there many "good"[8] modes hidden in the space of free ones. If we were able somehow to extract the subspace of those good modes out of the space of free ones and exclude the rest, we would obtain a solution that is truly independent of the grid. This would be like truncated SMS but with more modes and thus better results. These are still speculations, and exploring them will be our next step in improving the SMS method.

[8] By good modes, we mean those approximating the kernel modes well.

CHAPTER 4

Summary

In this thesis, we consider the analytic continuation problem which can be formulated as a Fredholm integral equation of first kind whose solution is known to be non-negative. We explain different earlier methods to solve this problem of which the most effective one is the non-negative Tikhonov method. This method gives good results in many cases and it is quite fast. However, it may introduce extra oscillations to the solution or set its values to zero when the original ones are small. Besides, it uses a parameter that needs to be tuned heuristically.

We move then to more computationally demanding methods; *Stochastic Sampling*. Assuming a multivariate normal distribution of the noise on the data and using the non-negativity of the solution, theses methods construct a probability distribution of the solutions which is truncated multivariate normal, and they use the mean as the optimal solution. The non-negativity constraint is essential for these methods, because without it, the probability distribution is a multivariate normal one whose mean is the same as its modal. But this is nothing but the generalized least squares solution which we already know to be a bad solution dominated by sawtooth noise. Using the constraint, however, the method shows promising results.

Earlier approaches typically sample the distribution using the Metropolis algorithm where updates are suggested on the solution's components directly. Since the components are highly correlated, the correlation times are high. They are so-high that a simulated annealing procedure is needed. To reduce correlation time, we propose a new method *Stochastic Mode Sampling*, where we sample the projection coefficients of the solution on right singular vectors (modes) of the kernel (modified by the data correlation matrix). These coefficients are statistically uncorrelated by construction, and we sample them using Gibbs sampling. The most tricky point in the new method is the coupling between the coefficients resulting from the non-negativity constrain.

Preliminary results using small and spare grids are surprisingly good! The solutions have good agreement with the original ones, and the method has small correlation times (one million correlated samples are often enough for a good estimate of the mean). However, as we start making the gird larger, the correlation time increases significantly. This is the result of the kernel being an increasing function of the grid coordinate. We solve the problem by rewriting the integral equation such that the kernel function becomes decaying. Moreover, we show that the results are independent of the specific modification.

Equipped with the modification technique, we are able to take larger uniform girds. Unfortunately, the results get worse instead of converging and by trying different nonuniform ones, the results change (some get better)! This is the result of the so-called free modes which change drastically from one grid to another. By removing those modes from the sampling, the results for different grids do actually converge and the correlation time is very small (few hundred thousands correlated samples are often enough). We call this variant of our method, *Truncated SMS*.

Although the results of Truncated SMS are grid-independent and it is fast in comparison to full SMS, the quality of its solution is not as good as some of the selected results of the full SMS. When the grid concentrates the free modes in the region where the solution is concentrated, the SMS solution is better than the truncated SMS one. We suggest a nonuniform grid that satisfies this condition, and we parametrize it such that it can be adapted to different cases. We provide a heuristic argument of why this type of grid gives better results, but the proposed grid is not unique and a full explanation is still required.

APPENDIX A

Green Functions and Analytic Continuation

In this appendix, we provide an introduction to Green functions and summarize their analytic properties. The goal is to provide the necessary background to understand the analytic continuation problem of Green functions from the physics point of view.

The mathematical definition of Green function of a differential equation is first introduced, followed by its application to one body Schrödinger equation. The concept of Green functions is then generalized to the many-body case and then further to imaginary-time. Finally, we conclude with a description of the analytic continuation problem.

A.1. Mathematical Definition

The concept of Green functions was first developed as a mathematical tool to solve inhomogeneous linear differential equations

$$Lu(x) = f(x) \tag{A.1}$$

where L is a linear differential operator and x stands for space and/or time variables. If the solution $G(x, x')$ for a point source at point x' is known

$$LG(x, x') = \delta(x - x') \tag{A.2}$$

then, due to the linearity of the equation, the solution for the source $f(x)$ reads

$$u(x) = \int dx' G(x, x') f(x') \tag{A.3}$$

$G(x, x')$ is called the *Green function* associated with differential equation (A.1) or equivalently the operator L.

A.2. One-Body Green Function

A.2.1. Time Independent

Let us apply the previous definition to the time-independent Schrödinger equation of a single particle

$$(z - H)\Psi(\mathbf{r}) = 0 \tag{A.4}$$

where z is complex parameter.[1] The associated Green function $G(x, x'; z) = G(\mathbf{r}, \mathbf{r}'; z)$ is then defined by

$$(z - H)\, G(\mathbf{r}, \mathbf{r}', z) = \delta(\mathbf{r} - \mathbf{r}') \tag{A.5}$$

The previous equation can equivalently be written in operator form

$$(z - H)G(z) = I \tag{A.6}$$

which allows us to define the *Green operator* (also known as *resolvent*) corresponding to H

$$G(z) \equiv (z - H)^{-1} \tag{A.7}$$

Using this operator, we can get Green functions in any basis by taking appropriate matrix elements in that basis. For example, the Green function defined in Eq. (A.5) is just the Green operator in configuration space.

[1] When the equation is satisfied, z is real and corresponds to the eigenenergys of the hermitian operator H.

Analytic Structure Now we turn to the analytic properties of this operator. Since H is hermitian, it has a complete set of orthonormal eigenstates $|n\rangle$ with real eigenenergies E_n. In general, the spectrum of H has both discrete part (corresponding to bound states) and continuous part (corresponding to scattering states).

$$H|n\rangle = E_n|n\rangle \tag{A.8}$$

$$\langle m|n\rangle = \delta_{mn} \tag{A.9}$$

$$\sum_n |n\rangle\langle n| \equiv \sum_{n'} |n'\rangle\langle n'| + \int dm\,|m\rangle\langle m'| = \mathbb{1} \tag{A.10}$$

where we use the convention that the sum over $|n'\rangle$ states is over the discrete spectrum only and the sum over $|n\rangle$ states is a sum over the discrete spectrum and an integral over the continuous spectrum (when they exist).

Using Eq. (A.10), the resolvent can be multiplied by the identity operator to get

$$G(z) = \frac{1}{z-H}\sum_n |n\rangle\langle n| = \sum_{n'} \frac{|n'\rangle\langle n'|}{z-E_{n'}} + \int \mathrm{d}m \frac{|m\rangle\langle m|}{z-E_m} \tag{A.11}$$

From the above expression, we expect $G(z)$ to be analytic[2] in the complex plane except on the part of the real axis where z matches one of the eigenenergies. The discrete part of the spectrum gives rise to simple poles while the continuous part gives rise to a branch cut. (see Fig. A.1)

$G(z)$, then, has two different values depending on whether the branch cut is approached from above or below. So we can define two functions of real parameter E

$$G^{\pm}(E) = \lim_{\epsilon\to 0^+} G(E \pm i\epsilon) = p.v.\left(\sum_n \frac{|n\rangle\langle n|}{E-E_n}\right) \mp i\pi \sum_n \delta(E-E_n)|n\rangle\langle n| \tag{A.12}$$

where the following identity was used

$$\lim_{\epsilon\to 0^+} \frac{1}{x \pm i\epsilon} = p.v.\left(\frac{1}{x}\right) \mp i\pi\delta(x) \tag{A.13}$$

Note that, due to the principal value, $G^{\pm}(E)$ are defined on the whole real axis as *distributions*.

The difference of $G^{\pm}(E)$ gives the discontinuity across the real axis (at the spectral values), which is purely imaginary:

$$\tilde{G}(E) \equiv G^+(E) - G^-(E) = -2\pi i \sum_n \delta(E-E_n)|n\rangle\langle n| \tag{A.14}$$

Taking the diagonal matrix elements in configuration space, we can relate it to the density of states per unit volume $\rho(\mathbf{r}, E)$

$$\tilde{G}(\mathbf{r},\mathbf{r};E) = -2\pi i \sum_n \delta(E-E_n)|\phi_n(\mathbf{r})|^2 = -2\pi i\rho(\mathbf{r},E) \tag{A.15}$$

[2] $G(z)$ is analytic if the matrix element $\langle\phi|G(z)|\chi\rangle$ is analytic for all $|\phi\rangle$ and $|\chi\rangle$

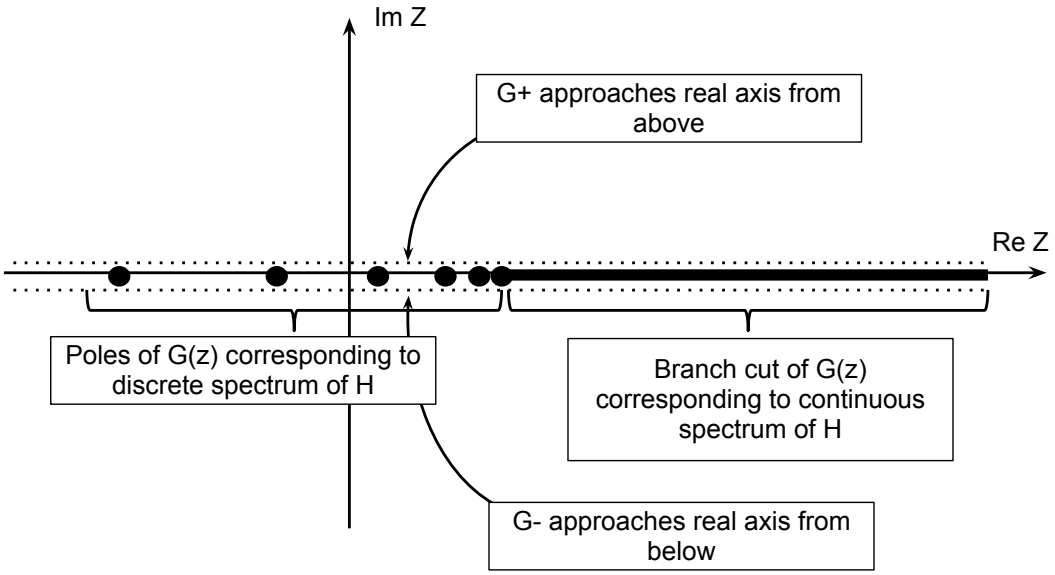

Fig. A.1.: Analytic structure of the Green operator in z complex plane

Finally, we relate $G(z)$ to $\tilde{G}(z)$ and $\rho(E)$:

$$\begin{aligned} G(z) &= \sum_n \frac{|n\rangle \langle n|}{z - E_n} = \int_{-\infty}^{+\infty} dE \sum_n \delta(E - E_n) \frac{|n\rangle \langle n|}{z - E} \\ &= \frac{-1}{2\pi i} \int_{-\infty}^{+\infty} dE \frac{\tilde{G}(E)}{z - E} = \int_{-\infty}^{+\infty} dE \frac{\rho(E)}{z - E} \end{aligned} \tag{A.16}$$

A.2.2. Time Dependent

Now we apply the definition to the time-dependent Schrödinger equation of a single particle

$$\left[i\hbar \frac{\partial}{\partial t} - H \right] \Psi(x) = 0 \tag{A.17}$$

The Green function associated with this equation $G(x, x') = G(\mathbf{r}, \mathbf{r}', t, t')$ satisfies:

$$\left[i\hbar \frac{\partial}{\partial t} - H \right] G(\mathbf{r}, \mathbf{r}', t, t') = \delta(\mathbf{r} - \mathbf{r}')\delta(t - t') \tag{A.18}$$

When H is time-independent, the system is invariant regarding time translations and G is only a function of the difference $t - t'$ so we can, without loss of generality, set $t' = 0$ and perform one Fourier transform in time

$$G(\mathbf{r}, \mathbf{r}', t) \equiv G(\mathbf{r}, \mathbf{r}', t, 0) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} d\omega e^{-i\omega t} G(\mathbf{r}, \mathbf{r}', \omega) \tag{A.19}$$

Substituting Eq. (A.19) in Eq. (A.18), we get

$$(\hbar\omega - H)G(\mathbf{r}, \mathbf{r}', \omega) = \delta(\mathbf{r} - \mathbf{r}') \tag{A.20}$$

By employing atomic units[3] where $\hbar = 1$ and comparing with Eq. (A.5), we see that z parameter of time-independent Green function represents (up to a multiplication factor) the frequency dependence of the time-dependent one.

As shown above, $G(w)$ (or equivalently $G(z)$) is not analytic on the whole real axis, which makes Eq. (A.19) not well-defined. To make it a well-defined integration, an integration path in the complex plane, infinitesimally close to the real axis, should be chosen. Two such paths are the ones corresponding to $G^{\pm}$ (see Fig. A.1) which leads to two different Green functions of time

$$G^{\pm}(t) = \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} e^{-i\omega t} G^{\pm}(\omega) \tag{A.21}$$

Note that when τ is positive, we need to close the integration path in the lower complex plane which makes $G^{-}(\tau)$ zero, while for τ negative, we need to close the integration path in the upper complex plane which makes $G^{+}(\tau)$ zero.

Now their difference (which is nothing but the Fourier transform of $\tilde{G}(w)$) relates to the time evolution operator

$$\begin{aligned} \tilde{G}(t) = G^{+}(t) - G^{-}(t) &= \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} e^{-i\omega t} \left[G^{+}(\omega) - G^{-}(\omega) \right] \\ &= \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} e^{-i\omega t} \tilde{G}(\omega) = -i \sum_{n} e^{-iE_n t} \left| n \right\rangle \left\langle n \right| \\ &= -ie^{-iHt} = -iU(t) \end{aligned} \tag{A.22}$$

In configuration space, we can write (restoring the other time index)

$$iG^{+}(\mathbf{r}, \mathbf{r}', t, t') = i\theta(t - t')\tilde{G}(\mathbf{r}, \mathbf{r}', t, t') = \theta(t - t') \left\langle \mathbf{r} \right| e^{-iH(t-t')} \left| \mathbf{r}' \right\rangle \tag{A.23}$$

$$-iG^{-}(\mathbf{r}, \mathbf{r}', t, t') = i\theta(t' - t)\tilde{G}(\mathbf{r}, \mathbf{r}', t, t') = \theta(t' - t) \left\langle \mathbf{r} \right| e^{-iH(t-t')} \left| \mathbf{r}' \right\rangle \tag{A.24}$$

Let us now verify that G^{+} is indeed a Green function, by substituting its formula in Eq. (A.18) and remembering that $\theta'(\tau) = \delta(\tau)$:

$$\begin{aligned} &\left[i\frac{\partial}{\partial t} - H \right] G^{+}(\mathbf{r}, \mathbf{r}', t, t') = \left[i\frac{\partial}{\partial t} - H \right] (-i)\theta(t - t') \left\langle \mathbf{r} \right| e^{-iH(t-t')} \left| \mathbf{r}' \right\rangle = \\ &\left[i\frac{\partial}{\partial t} - H \right] (-i)\theta(t - t') \sum_{n} e^{iE_n t'} \phi_n^*(\mathbf{r}') e^{-iE_n t} \phi_n(\mathbf{r}) = \\ &\sum_{n} e^{iE_n t'} \phi_n^*(\mathbf{r}') \left\{ \cancel{-i\theta(t - t') \left[i\frac{\partial}{\partial t} - H \right] e^{-iE_n t} \phi_n(\mathbf{r})} + \delta(t - t') e^{-iE_n t} \phi_n(\mathbf{r}) \right\} \\ &= \delta(t - t') \sum_{n} \phi_n^*(\mathbf{r}') \phi_n(\mathbf{r}) = \delta(t - t') \sum_{n} \left\langle r | n \right\rangle \left\langle n | r' \right\rangle = \delta(t - t')\delta(\mathbf{r} - \mathbf{r}') \end{aligned} \tag{A.25}$$

[3] From now on, we will work in atomic units.

where the crossed term is zero because $e^{-iE_n t}\phi_n(\mathbf{r})$ is a solution of the time-dependent Schrödinger equation.

Similarly, we can check that G^- is Green function. Note, however, that $\tilde{G}(\tau)$ satisfies Eq. (A.17) but not Eq. (A.18) and thus it is not a Green function according to the mathematical definition (that is because it lacks the necessary jump at equal times leading to a delta function in time).

A.3. Many-Body Green Function

A.3.1. Real Time

Now we generalize the concept of Green function to the many-body case. The *retarded* and *advanced* Green functions of a many-body system are defined respectively:

$$G^R(\mathbf{r},\mathbf{r}',t,t') = -i\theta(t-t')\left\langle \left[\hat{\psi}(\mathbf{r},t),\hat{\psi}^\dagger(\mathbf{r}',t')\right]_\pm \right\rangle \tag{A.26}$$

$$G^A(\mathbf{r},\mathbf{r}',t,t') = i\theta(t'-t)\left\langle \left[\hat{\psi}(\mathbf{r},t),\hat{\psi}^\dagger(\mathbf{r}',t')\right]_\pm \right\rangle \tag{A.27}$$

where $\hat{\psi}$ and $\hat{\psi}^\dagger$ are the field operators in the Heisenberg picture, $\langle ... \rangle$ is an expectation value to be defined next, $[...]_+$ is the anti-commutator (for fermions) and $[...]_-$ is the commutator (for bosons)[4]

Zero and Finite Temperature There are two classes of Green functions, zero temperature and finite temperature. The previous definitions are valid for both and which is which depends on the interpretation of the expectation value.
At zero temperature, the system stays in its ground state $|\Phi_0\rangle$ and all the information we need about an operator $\hat{O}$ at zero temperature is contained in the following expectation value

$$\left\langle \hat{O} \right\rangle = \frac{\langle \Phi_0 | \hat{O} | \Phi_0 \rangle}{\langle \Phi_0 | \Phi_0 \rangle} \tag{A.28}$$

On the other hand, at finite temperature the system fluctuates between different energy levels and the probability of finding the system at a specific level depends on the external constrains and it is determined by means of statistical mechanics. In this case, the expectation value should take into account both quantum and statistical averages.

We are interested in the so called *Grand Canonical Ensemble*, where the system is allowed to exchange not only energy, but also particles with the surrounding while kept at fixed

[4] We will address both the fermoinic and the bosonic case in most formulas simultaneously, where the upper sign refers to ferminons and the lower sign to bosons.

temperature T and chemical potential μ. With these constrains, the probability of finding the system at energy level E with N particles is proportional to $e^{-(E-\mu N)/(kT)}$.

In this ensemble the finite-temperature expectation value of the operator $\hat{O}$ is defined as

$$\left\langle \hat{O} \right\rangle = \frac{Tr\left[e^{-\beta(\hat{H}-\mu\hat{N})}\hat{O}\right]}{Tr\left[e^{-\beta(\hat{H}-\mu\hat{N})}\right]} = \frac{\sum_n \langle n|\, e^{-\beta(\hat{H}-\mu\hat{N})}\hat{O}\,|n\rangle}{\sum_n \langle n|\, e^{-\beta(\hat{H}-\mu\hat{N})}\,|n\rangle} \tag{A.29}$$

where the sum is over all eigenstates of $\hat{H}$ in Fock space (i.e. the sum runs over states with different particle numbers), $\hat{N}$ is the number of particles operator and $\beta = 1/kT$ is called the inverse temperature.

Note 1 In the grand canonical ensemble, it is convenient to modify the definition of Heisenberg operators as following

$$\hat{O}(t) \equiv e^{i(\hat{H}-\mu\hat{N})t}\hat{O}e^{-i(\hat{H}-\mu\hat{N})t} \tag{A.30}$$

This has the advantage of using the same operator $\hat{H} - \mu\hat{N}$ for both time evolution and thermal averaging. It will produce the same results as if we had used $\hat{H}$ for time evolution instead, as long as the Hamiltonian $\hat{H}$ preserves the number of particle (which is assumed to be the case) and the operator $\hat{O}$ also does not change the number of particles (which is the case for combinations of paired $\hat{\psi}$ and $\hat{\psi}^\dagger$). The first property means that $\hat{H}$ commutes with $\hat{N}$, so the exponentials can be factorized into two terms $e^{\pm i\hat{H}t}$ and $e^{\mp i\mu\hat{N}t}$. The second property means that $e^{\mp i\mu\hat{N}t}$ commute with $\hat{O}$, so we can combine them getting the unity operator and we are back to the original definition of Heisenberg operators.

With this modification and since now only $\hat{H} - \mu\hat{N}$ appears, we will shorten the notation in the grand canonical ensemble and use $\hat{H}$ to actually denote $\hat{H} - \mu\hat{N}$ and E_n to denote $E_n - \mu N$. This allows us to handle both the zero and finite temperature cases with the same notation.

Note 2 When $T \to 0$, then $\beta \to \infty$ and only the state with lowest energy survive the exponentially damping factor and we are actually back to the zero temperature definition. This assumes that the ground state is non-degenerate, otherwise we have a linear combination of the degenerate ground states.

Other Bases and Frequency Domain The previous definition of the Green function was stated in configuration space but it can be equally defined in any other single-particle basis

$$G^{R/A}(\kappa, \kappa', t, t') = -i\theta(t-t')\left\langle \left[\hat{c_\kappa}(t), \hat{c_{\kappa'}}^\dagger(t')\right]_\pm \right\rangle \tag{A.31}$$

and the relation between the two is

$$G^{R/A}(\mathbf{r}, \mathbf{r}', t, t') = \sum_\kappa \sum_{\kappa'} \phi_\kappa(\mathbf{r})\phi^*_{\kappa'}(\mathbf{r}')G^R(\kappa, \kappa', t, t') \tag{A.32}$$

where $\phi_\kappa(\mathbf{r})$ are the basis wavefunctions.

We can also switch from time to frequency using the Fourier transform

$$G^{R/A}(\kappa,\kappa',t) \equiv G^{R/A}(\kappa,\kappa',t,0) = \int \frac{d\omega}{2\pi}\, e^{-i\omega t} G^{R/A}(\kappa,\kappa',\omega) \tag{A.33}$$

$$G^{R/A}(\kappa,\kappa',\omega) = \int dt\, e^{i\omega t} G^{R/A}(\kappa,\kappa',t) \tag{A.34}$$

where the Hamiltonian is assumed to be time-independent and so only time differences matter.

Relation to One-Body Green Functions Do the many-body Green functions comply, with the mathematical definition? i.e. are $G^{R/A}$ Green functions of some differential equation?
In general, No! Only when the particles are **non-interacting**, $G^{R/A}$ are the Green functions of an equation; Namely, the Schrödinger equation with the single particle Hamiltonian $\hat{h_i}$, where the full system Hamiltonian is $H = \sum_{i=1}^{N} h_i$.
Let us check this. In the non-interacting case, the Hamiltonian is diagonal in some single-particle basis $\hat{H} = \sum_n E_n \hat{c}_n^\dagger \hat{c}_n$, then the time dependence of the creation/annihilation operators is trivial:

$$c_n(t) = e^{-iE_n t}\hat{c}_n,\ \hat{c}_n^\dagger(t) = e^{+iE_n t}\hat{c}_n^\dagger \tag{A.35}$$

and we can write

$$\begin{aligned}
&\left\langle \left[\hat{\psi}(\mathbf{r},t), \hat{\psi}^\dagger(\mathbf{r}',t')\right]_\pm \right\rangle = \\
&\sum_n \sum_{n'} e^{iE_{n'}t'} \phi_{n'}^*(\mathbf{r}') e^{-iE_n t} \phi_n(\mathbf{r}) \left\langle \left[\hat{c}_n, \hat{c}_{n'}^\dagger\right]_\pm \right\rangle = \\
&\sum_n \sum_{n'} e^{iE_{n'}t'} \phi_{n'}^*(\mathbf{r}') e^{-iE_n t} \phi_n(\mathbf{r}) \left\langle \delta_{nn'} \right\rangle = \\
&\sum_n e^{iE_n t'} \phi_n^*(\mathbf{r}') e^{-iE_n t} \phi_n(\mathbf{r}) = \left\langle \mathbf{r} \left| e^{-ih_i(t-t')} \right| \mathbf{r}' \right\rangle
\end{aligned} \tag{A.36}$$

Comparing this with Eq. (A.23) and Eq. (A.24), we see that for the non-interacting case:

$$G^R = G^+,\ G^A = G^- \tag{A.37}$$

Causal Green Function A third Green function can also be defined[5] and it is called the *causal* Green function

$$G^C(\mathbf{r},\mathbf{r}',t,t') = -i \left\langle T\left\{\hat{\psi}(\mathbf{r},t), \hat{\psi}^\dagger(\mathbf{r}',t')\right\}\right\rangle \tag{A.38}$$

[5] In the non-interacting case, it is also a Green function of single-particle Schrödinger equation.

where T is the time-ordering operator which orders its argument in chronological order from right to left

$$T\left\{\hat{\psi}(\mathbf{r},t),\hat{\psi}^{\dagger}(\mathbf{r}',t')\right\}=\begin{cases}\hat{\psi}(\mathbf{r},t)\hat{\psi}^{\dagger}(\mathbf{r}',t') & \text{if } t'<t\\ \mp\hat{\psi}^{\dagger}(\mathbf{r}',t')\hat{\psi}(\mathbf{r},t) & \text{if } t'>t\end{cases} \tag{A.39}$$

A.3.2. Imaginary Time

In the imaginary-time formalism of finite temperature Green functions, one replaces the time parameter using the substitution $it\to\tau$. There are two motivations for this change of variables. From the analytic point of view, the simultanous appearance of $\hat{H}$ in the exponential, once with the real prefactor β and another with the imaginary prefactor it, suggests that making both prefactors real would make things more uniform and easier to handle. From the numerical point of view, it turned out that calculating retarded Green function at finite temperature directly is faced with technical problems which are avoided when we go to imaginary time.

The imaginary-time Green function (also called Matsubara Green function) is defined as

$$\mathcal{G}(\kappa,\kappa',\tau,\tau')\equiv-\left\langle T\left\{\hat{c}_{\kappa}(\tau),\hat{c}^{\dagger}_{\kappa'}(\tau')\right\}\right\rangle \tag{A.40}$$

where τ,τ' are real numbers in the range $[-\beta,+\beta]$, T is the time ordering operator defined in Eq. (A.39) and the operators are defined in the imaginary-time Heisenberg picture where[6]

$$\hat{O}(\tau)\equiv e^{\hat{H}\tau}\hat{O}e^{-\hat{H}\tau}$$

Periodicity and Frequency Domain First, as usual we assume the Hamiltonian is time independent and so only time differences matter and we can set the second time parameter to zero

$$\mathcal{G}(\kappa,\kappa',\tau)\equiv\mathcal{G}(\kappa,\kappa',\tau,0) \tag{A.41}$$

Now we state without proof that the imaginary-time Green function has the following anti-periodic (for fermions) or periodic (for bosons) property

$$\mathcal{G}(\kappa,\kappa',\tau+\beta)=\mp\mathcal{G}(\kappa,\kappa',\tau)\text{ for }-\beta\leq\tau<0 \tag{A.42}$$

or equivalently

$$\mathcal{G}(\kappa,\kappa',\tau-\beta)=\mp\mathcal{G}(\kappa,\kappa',\tau)\text{ for }0<\tau\leq+\beta \tag{A.43}$$

Note that these relations hold only inside the interval $[-\beta,+\beta]$. However, they can be imposed outside the interval *by definition*[7].

[6] Remember that our convention of $\hat{H}$ representing $\hat{H}-\mu\hat{N}$ still holds.

[7] Remember that $\mathcal{G}$ was originally defined only inside the interval $[-\beta,+\beta]$ and we can extend it outside the interval by repeating its values periodically.

Using the (anti-)periodicity, imaginary-time Green function can be expanded as a Fourier series

$$\mathcal{G}(\kappa,\kappa',\tau) = \frac{1}{\beta}\sum_{\omega_n} e^{-i\omega_n\tau}\mathcal{G}(\kappa,\kappa',i\omega_n) \tag{A.44}$$

$$\mathcal{G}(\kappa,\kappa',i\omega_n) = \int_0^\beta d\tau e^{i\omega_n\tau}\mathcal{G}(\kappa,\kappa',\tau) \tag{A.45}$$

where ω_n are called *Matsubara Frequencies*. They are the only frequencies where $\mathcal{G}(i\omega_n)$ is defined and are given by

$$\begin{aligned}\omega_n \equiv& \frac{(2n+1)\pi}{\beta} \text{ (Fermions)}\\ \omega_n \equiv& \frac{2n\pi}{\beta} \text{ (Bosons)}\end{aligned} \tag{A.46}$$

where n is any integer number.

A.4. Analytic Properties and Analytic Continuation

From the definitions of Matsubara and causal Green functions, we can already establish a connection between the two in the time domain

$$\mathcal{G}(\kappa,\kappa',\tau) = -iG^C(\kappa,\kappa',-i\tau) \tag{A.47}$$

The relation between the retarded and advanced Green functions and Matsubara Green function is a bit trickier and to see it we need to go to the so called *Lehmann Representation* in frequency domain.

Lehmann Representation We insert a complete set of eigenstates (complete in Fock space, so no restriction on particles number) in the diagonal elements of Green function

$$\begin{aligned}G^R(\kappa,t) &\equiv G^R(\kappa,\kappa,t,0)\\ &= -i\theta(t)\frac{1}{Z}\Big\{\sum_n\sum_m \langle n|\, e^{-\beta\hat{H}}e^{i\hat{H}t}\hat{c_\kappa}e^{-i\hat{H}t}\,|m\rangle\,\langle m|\,\hat{c_\kappa}^\dagger\,|n\rangle\\ &\qquad \pm\sum_n\sum_m \langle n|\, e^{-\beta\hat{H}}\hat{c_\kappa}^\dagger\,|m\rangle\,\langle m|\, e^{i\hat{H}t}\hat{c_\kappa}e^{-i\hat{H}t}\,|n\rangle\Big\}\\ &= -i\theta(t-t')\frac{1}{Z}\Big\{\sum_n\sum_m e^{-\beta E_n}e^{i(E_n-E_m)t}|\,\langle m|\,c_\kappa^\dagger\,|n\rangle\,|^2\\ &\qquad \pm\sum_n\sum_m e^{-\beta E_n}e^{i(E_m-E_n)t}|\,\langle n|\,c_\kappa^\dagger\,|m\rangle\,|^2\Big\}\\ &= -i\theta(t)\frac{1}{Z}\sum_n\sum_m\{e^{-\beta E_n}\pm e^{-\beta E_m}\}e^{i(E_n-E_m)t}|\,\langle m|\,c_\kappa^\dagger\,|n\rangle\,|^2\end{aligned} \tag{A.48}$$

where $Z = \sum_n \langle n| e^{-\beta \hat{H}} |n\rangle$.

Similarly, for the other Green functions:

$$G^A(\kappa, t) = i\theta(-t)\frac{1}{Z}\sum_n \sum_m \{e^{-\beta E_n} \pm e^{-\beta E_m}\} e^{i(E_n - E_m)t} |\langle m| c_\kappa^\dagger |n\rangle|^2 \tag{A.49}$$

$$\mathcal{G}(\kappa, \tau) = -\frac{1}{Z}\sum_n \sum_m \{\theta(t) e^{-\beta E_n} \pm \theta(-t) e^{-\beta E_m}\} e^{(E_n - E_m)\tau} |\langle m| c_\kappa^\dagger |n\rangle|^2 \tag{A.50}$$

These are Lehmann representations in time domain. Now we go to the frequency domain. For real-time functions, we use Fourier transform Eq. (A.34)

$$\begin{aligned} G^R(\kappa, \omega) &= \int_{-\infty}^{+\infty} e^{i\omega t} G^R(\kappa, t) dt \\ &= \frac{-i}{Z}\sum_n \sum_m \{e^{-\beta E_n} \pm e^{-\beta E_m}\} |\langle m| c_\kappa^\dagger |n\rangle|^2 \int_{-\infty}^{+\infty} \theta(t) e^{i(\omega + E_n - E_m)t} dt \end{aligned} \tag{A.51}$$

But

$$\int_{-\infty}^{+\infty} \theta(t) e^{i(\omega + E_n - E_m)t} dt = \left[\frac{e^{i(\omega + E_n - E_m)t}}{i(\omega + E_n - E_m)}\right]_{t=0}^{t \to \infty} \tag{A.52}$$

where the upper limit is oscillating and has no specific value. But if we add an infinitesimally small positive imaginary part $\omega \to \omega + i\epsilon$, then the exponent becomes negative and the upper limit converges to zero. So we get

$$G^R(\kappa, \omega) = \frac{1}{Z}\sum_n \sum_m \{e^{-\beta E_n} \pm e^{-\beta E_m}\} \frac{|\langle m| c_\kappa^\dagger |n\rangle|^2}{\omega + E_n - E_m + i\epsilon} \tag{A.53}$$

Similarly for the advanced Green functions:

$$G^A(\kappa, \omega) = \frac{1}{Z}\sum_n \sum_m \{e^{-\beta E_n} \pm e^{-\beta E_m}\} \frac{|\langle m| c_\kappa^\dagger |n\rangle|^2}{\omega + E_n - E_m - i\epsilon} \tag{A.54}$$

For the imaginary-time Green function, we use the Fourier series Eq. (A.45)

$$\begin{aligned} \mathcal{G}(\kappa, i\omega_n) &= -\frac{1}{Z}\sum_n \sum_m e^{-\beta E_n} |\langle m| c_\kappa^\dagger |n\rangle|^2 \int_0^\beta d\tau e^{(i\omega_n + E_n - E_m)\tau} \\ &= -\frac{1}{Z}\sum_n \sum_m e^{-\beta E_n} |\langle m| c_\kappa^\dagger |n\rangle|^2 \frac{e^{(i\omega_n + E_n - E_m)\beta} - 1}{i\omega_n + E_n - E_m} \\ &= \frac{1}{Z}\sum_n \sum_m \{e^{-\beta E_n} \pm e^{-\beta E_m}\} \frac{|\langle m| c_\kappa^\dagger |n\rangle|^2}{i\omega_n + E_n - E_m} \end{aligned} \tag{A.55}$$

where $e^{i\omega_n \beta} = \mp 1$ using Eq. (A.46).

By comparing Lehmann representations Eqs. (A.53), (A.54), and (A.55), we see that we can switch between them using the substitutions

$$\begin{aligned} \text{Matsubara} &\longleftrightarrow \text{Retarded} \\ i\omega_n &\longleftrightarrow \omega + i\epsilon \end{aligned} \tag{A.56}$$

$$\begin{aligned} \text{Matsubara} &\longleftrightarrow \text{Advanced} \\ i\omega_n &\longleftrightarrow \omega - i\epsilon \end{aligned} \tag{A.57}$$

Spectral Function We now introduce the spectral function

$$A(\kappa,\omega) \equiv \frac{1}{Z}\sum_n\sum_m \{e^{-\beta E_n} \pm e^{-\beta E_m}\} |\langle m| c_\kappa^\dagger |n\rangle|^2 \delta(\omega + E_n - E_m) \tag{A.58}$$

this allows us to write the Lehmann representations in a concise form

$$G^R(\kappa,\omega) = \int d\omega' \frac{A(\kappa,\omega')}{(\omega + i\epsilon) - \omega'} \tag{A.59}$$

$$G^A(\kappa,\omega) = \int d\omega' \frac{A(\kappa,\omega')}{(\omega - i\epsilon) - \omega'} \tag{A.60}$$

$$\mathcal{G}(\kappa,i\omega_n) = \int d\omega' \frac{A(\kappa,\omega')}{i\omega_n - \omega'} \tag{A.61}$$

It is clear that the three functions are different faces of the same coin; a single function defined in the whole complex plane of frequency [8]

$$G(\kappa,z) = \int d\omega' \frac{A(\kappa,\omega')}{z - \omega'} \tag{A.62}$$

where $\mathcal{G}(\kappa,i\omega_n)$ are its values on the imaginary axis while $G^R(\kappa,\omega)$ and $G^A(\kappa,\omega)$ are its values slightly above and slightly below the real axis, respectively (see Fig. A.2).

We can read from Eq. (A.62) the analytic properties of G. On the part of the real axis where A is discrete i.e. a sum of delta functions at separate points, G has simple poles at those points. On the other hand, when A is continuous, which is the usually the case, we have a branch cut on the real axis. Compare this with the one-body case (see Fig. A.1).

The spectral function has two important properties :

1. Non-Negativity[9]: $A(\kappa,\omega) \geq 0$ (for fermions) , $A(\kappa,\omega)/\omega \geq 0$ (for bosons).
2. The sum rule[10]: $\int d\omega A(\kappa,\omega) = 1$.

[8] Note the similarity with the one-body case Eq. (A.16)

[9] This can be checked readily from the definition of the spectral function. For fermions, all terms are positive. For bosons, when $\omega > 0$ then $E_m > E_n$ so $(e^{-\beta E_n} - e^{-\beta E_m}) > 0$ and vice versa.

[10] This can be checked by plugging the definition in the integral, noticing that the integral of a delta function is one and employing the commutation/anti-commutation relations.

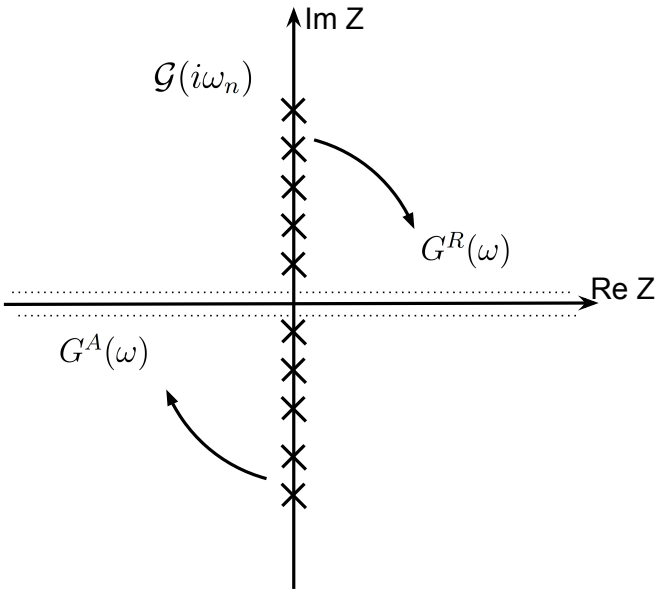

Fig. A.2.: Analytic Structure of $G(z)$. The Matsubara Green function $\mathcal{G}(i\omega_n)$ is defined on the imaginary axis at Matsubara frequencies and can be analytically continued in the upper half-plane to get the retarded Green function $G^R(\omega)$ or in the lower half-plane to get the advanced Green function $G^A(\omega)$.

Analytic Continuation Finally we come to the analytic continuation problem. The term analytic continuation refers, in general, to the process of obtaining the values of a function on some axis in the complex plane knowing its values on another axis. In our case, it is obtaining $G^{R/A}(\omega)$ from $\mathcal{G}(i\omega_n)$ (see Fig. A.2). However, the retarded and advanced Green functions are themselves not very interesting but rather the spectral function because many dynamical properties of the system are directly related to the spectral function. On the other hand, Quantum Monte Carlo (QMC) simulations provide Green function values at Matsubara frequencies $\mathcal{G}(i\omega_n)$ or imaginary-time points $\mathcal{G}(\tau)$.

We have already established the relation between $\mathcal{G}(i\omega_n)$ and $A(\omega)$ in Eq. (A.61) and to obtain the relation between $\mathcal{G}(\tau)$ and $A(\omega)$, we apply the inverse Fourier transform Eq. (A.44) to Eq. (A.61)

$$\mathcal{G}(\tau) = \frac{1}{\beta}\sum_{\omega_n} e^{-i\omega_n\tau}\mathcal{G}(i\omega_n) = \int d\omega A(\omega)\frac{1}{\beta}\sum_{\omega_n}\frac{e^{-i\omega_n\tau}}{i\omega_n - \omega} \tag{A.63}$$

Summation over Matsubara frequencies is done by applying the following trick

$$\frac{1}{\beta}\sum_{\omega_n} g(i\omega_n) = \frac{1}{\beta}\sum_{z_0\in \text{ poles of } h(z)} \text{Res } g(z_0)h(z_0) = \frac{1}{2\pi i\beta}\oint_{C1} dz\, g(z)h(z) \tag{A.64}$$

where $h(z)$ is an auxiliary function that has poles at Matsubara frequencies and residues

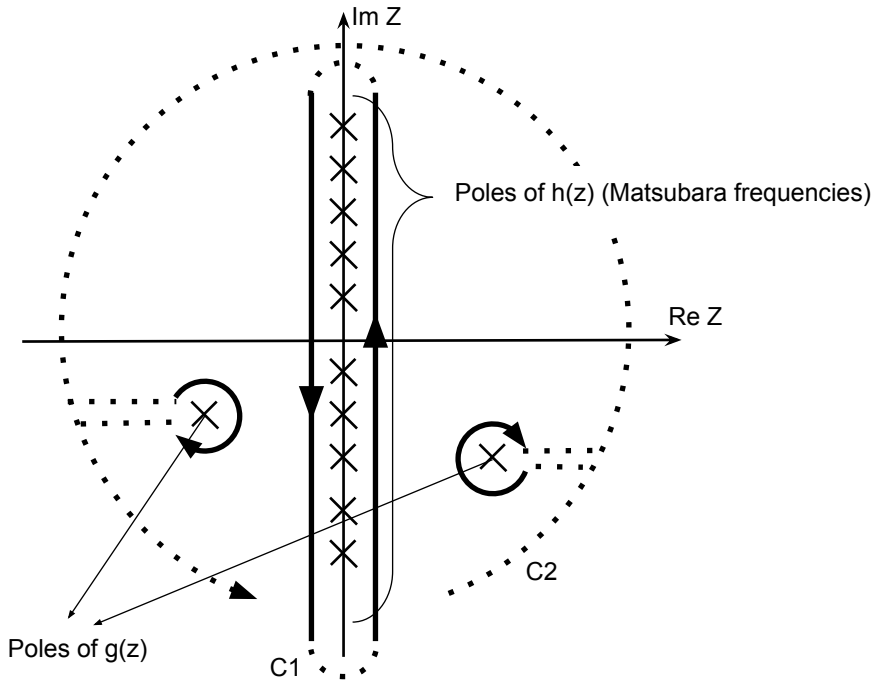

Fig. A.3.: Evaluating Matsubara frequencies summation using contour integral.

of one. Two such auxiliary functions are

$$h_1(z) = \frac{\beta}{1 \pm e^{-\beta z}},\ h_2(z) = \frac{-\beta}{1 \pm e^{\beta z}} \tag{A.65}$$

If g(z) has simple poles away from the imaginary axis, we can deform the integration path $C1$ into $C2$ (see Fig. A.3)

$$\oint_{C1} dz\ g(z)h(z) = \oint_{C2} dz\ g(z)h(z) \tag{A.66}$$

Assuming that $g(z)h(z)$ decays sufficiently at large z, contributions from the infinite circles of $C2$ vanish and the poles of $g(z)$ can be isolated

$$\frac{1}{\beta}\sum_{\omega_n} g(i\omega_n) = \frac{1}{2\pi i\beta}\oint_{C2} dz\ g(z)h(z) = -\frac{1}{\beta}\sum_{z_0 \in \text{ poles of } g(z)} \text{Res } g(z_0)h(z_0) \tag{A.67}$$

Now back to Eq. (A.63), we have $g(z) = e^{-iz\tau}/(z-\omega)$. For $\tau > 0$, we should choose $h_1(z)$ as an auxiliary function to balance the divergence in $g(z)$ in the left half plane. This gives us

$$\mathcal{G}(\tau) = -\int d\omega \frac{e^{-\tau\omega}}{1 \pm e^{-\beta\omega}} A(\omega)\ (\text{for } \tau > 0) \tag{A.68}$$

While for $\tau < 0$, we should choose $h_2(z)$ as an auxiliary function to balance the divergence in $g(z)$ in the right half plane. This gives us

$$\mathcal{G}(\tau) = \int d\omega \frac{e^{-\tau\omega}}{1 \pm e^{\beta\omega}} A(\omega) \text{ (for } \tau < 0) \tag{A.69}$$

the first relation is the interesting one because QMC simulations are done for positive times.

Summery Analytic continuation of Green function is the problem of finding spectral function values using Green function values either at Matsubara frequencies or at positive imaginary times using the inverse of following relations

$$\mathcal{G}(i\omega_n) = \int d\omega \ \frac{1}{i\omega_n - \omega} \quad A(\omega) \tag{A.70}$$

$$\mathcal{G}(\tau) = \int d\omega \ \frac{-e^{-\tau\omega}}{1 \pm \ e^{-\beta\omega}} \ A(\omega) \tag{A.71}$$

See Refs. [18, 19, 20, 21, 22] for further information on Green functions and their analytic properties.

APPENDIX B

Sampling Truncated Univariate Normal Distribution

A truncated univariate normal distribution $\mathcal{TN}(\mu, \sigma^2, a, b)$ is the probability distribution of a normally distributed random variable $\mathcal{N}(\mu, \sigma^2)$ whose values are bounded to the interval $[a, b]$. Regarding sampling this distribution, it is sufficient to focus on the standard distribution only, i.e. $\mu = 0$ and $\sigma = 1$ because a random variable r drawn form a general distribution $\mathcal{TN}(\mu, \sigma^2, a, b)$ is related by the transformation $r = \mu + \sigma * r'$ to a random variable r' drawn form the standard one $\mathcal{TN}(0, 1, a', b')$ where $a' = (a - \mu)/\sigma$ and $b' = (b - \mu)/\sigma$.

We describe here an efficient sampling algorithm based on Ref. [23]. The algorithm uses the accept-reject method heavily, so let us review it. If we have a probability distribution $f(x)$ that is hard to sample, but we can sample a closely related distribution $g(x)$ where $Mg(x)$ is always above $f(x)$ for some constant M, then we can sample $f(x)$ by sampling $g(x)$ and accepting samples with probability $f(x)/(Mg(x))$. The efficiency of this method is measured by its acceptance rate which is the average acceptance probability

$$\int dx g(x)(f(x)/Mg(x)) = 1/M \ . \tag{B.1}$$

This implies that M should be as small as possible while still $Mg(x) \geq f(x)$. Therefore, the best value of M is the maximum of $f(x)/g(x)$.

Verifying that the accept-reject method gives the desired distribution is easy. The probability of obtaining a sample in the interval xdx in a single run is $g(x)(f(x)/Mg(x))dx = f(x)/Mdx$. In a large set of samples drawn using this method, the number of samples in that interval is proportional to $f(x)/Mdx$, while the total number of samples is proportional to the acceptance rate, $1/M$. This mean that samples are distributed according to

$f(x)$. Sec. II.3 of Ref. [24] provides rigorous mathematical treatment. Also see Sec. 7.3.6 of Ref. [25] for a geometrical explanation of the method.

Let us first discuss sampling left-truncated normal distribution, i.e. $a \geq 0$ and $b = \infty$. This will be used later as a sub-algorithm within the main one. A straightforward way of sampling is to propose samples from the normal distribution and reject the ones that are smaller than a. The acceptance rate of this method is simply I_a/I where

$$I_a \coloneqq Pr(x \geq a) = \int_a^{+\infty} e^{-x^2/2} dx = \frac{1}{2} \operatorname{erfc}(\frac{a}{\sqrt{2}}) \,, \tag{B.2}$$

$$I \coloneqq \int_{-\infty}^{+\infty} e^{-x^2/2} dx = \sqrt{2\pi} \,, \tag{B.3}$$

and erfc is the complementary error function. For small a, the acceptance rate is around 0.5, but it approaches zero very quickly for large a, so we need a more efficient method.

We use the accept-reject method with $f(x) = e^{-x^2/2}/I_a$, the desired left-truncated normal distribution, and $g(x) = \alpha e^{-\alpha(x-a)}$, the generalized exponential distribution in the interval $[a, \infty]$. The best value of M, as discussed earlier, should be chosen as the maximum of the function

$$\frac{f(x)}{g(x)} = \frac{1}{\alpha I_a} e^{-x^2/2-\alpha(x-a)} = \frac{1}{\alpha I_a} e^{-(x-\alpha)^2/2} e^{-\alpha a+\alpha^2/2} \,. \tag{B.4}$$

This function is a Gaussian of mean α and maximum $M = e^{-\alpha a+\alpha^2/2}/(\alpha I_a)$ which lies within the interval $[a, \infty]$ if $\alpha \geq a$.

The parameter α is chosen such that it maximizes the acceptance rate $1/M = \alpha I_a e^{\alpha a-\alpha^2/2}$. Setting the derivative of the acceptance rate to zero gives the best value $\alpha^\star = (a + \sqrt{a^2+4})/2$ which corresponds to an acceptance rate of $\alpha^\star I_a e^{(\alpha^\star a-1)/2}$. As a function of a, the worst case acceptance rate of this method is 0.5 for $a = 0$ and it increases rapidly as a gets larger.

Now we are back to sampling the general doubly-truncated normal distribution. We distinguish between three cases depending on the positioning of the truncation interval:

- **Case 1:** $a < 0 < b$ (mean is within the interval)

  If the interval is wide enough, a reasonable method is to repeatedly sample the normal distribution and reject samples lying outside the interval $[a, b]$. The acceptance rate of this method is $(I_a - I_b)/I$ where I_a and I are defined in Eq. B.2 and I_b is defined similarly.

  When the interval is narrow, it is better to start from the numbers within the interval. We use the accept-reject method with with $f(x) = e^{-x^2/2}/(I_a - I_b)$, the desired truncated normal distribution, and $g(x) = 1/(b-a)$, the uniform distribution in the interval $[a, b]$. In this case, $M = (b-a)/(I_a - I_b)$ and the acceptance rate is $(I_a - I_b)/(b-a)$.

  Comparing the acceptance rates of the two methods, we see that the second one is more efficient when the interval is narrow, specifically when $b - a < \sqrt{2\pi}$.

- **Case 2:** $0 \leq a < b$ (interval is on the left tail)

  If the interval is wide enough, we use the previously described efficient algorithm of sampling the left-truncated normal distribution to get samples greater than a. Then we accept only samples smaller than b. The acceptance rate of this method is $(I_b - I_a)/I_a$. Combining it with the acceptance rate of the sub-algorithm, we get a total acceptance rate of $(I_b - I_a)\alpha^\star e^{(\alpha^\star a - 1)/2}$.

  When the interval is narrow, we use accept-reject method with $f(x) = e^{-x^2/2}/(I_a - I_b)$, the desired truncated normal distribution, and $g(x) = 1/(b-a)$, the uniform distribution in the interval $[a, b]$. In this case, $M = (b-a)/(I_a - I_b)e^{-a^2/2}$ and the acceptance rate is $e^{a^2/2}(I_a - I_b)/(b-a)$.

  Comparing the acceptance rates of the two methods, we see that the second one is more efficient when the interval is narrow, specifically when $b - a < \frac{2\sqrt{e}}{a+\sqrt{a^2+4}}\, e^{\frac{a^2 - a\sqrt{a^2+4}}{4}}$.

- **Case 3:** $a < b \leq 0$ (interval is on the right tail)
  Draw a positive sample from the interval $[-b, -a]$ as descried in the case 2, then negate the result.

Implementation

We provide here a python implementation of the previously described method. The main function is *tnorm*; all other functions are auxiliary.

```
from math import *
from numpy import random
from scipy.stats import expon
from scipy.stats import norm

#Left-truncated standard normal
def ltnorm_std(a):
    alpha=0.5*(a+sqrt(a**2+4))
    i=0
    while(True):
        #This procedure takes the scale parameter of the exponential
        #distribution as the second argument. This is simply the inverse
        #of the rate parameter 'alpha' used in text.
        z=expon.rvs(a,1.0/alpha)
        rho=exp(-0.5*(z-alpha)**2)
        u=random.uniform(0,1)
        if(u<=rho):
            return z

#Case1: use it only when a*b<0. It works when either a or b is infinity.
def tuvnorm1(a,b):
    if((b-a)>=sqrt(2*pi)):
        while(True):
            r=norm.rvs(0,1)
            if((r>=a)and(r<=b)):
```

```
                return r
        else:
            while(True):
                z = random.uniform(a,b)
                rho = exp(-0.5*z**2)
                u=random.uniform(0,1)
                if(u<=rho):
                    return z

#Case2: use it only when 0<a<b. It works when b is infinity.
def tuvnorm2(a,b):
    temp=sqrt(a**2+4)
    if(b>a+(  2*exp(0.5+0.25*a*(a-temp))/(a+temp))):
        while(True):
            r=ltnorm(a)

            if(r<=b):
                return r
    else:
        while(True):
            z = random.uniform(a,b)
            rho = exp(0.5*(a**2-z**2))
            u=random.uniform(0,1)
            if(u<=rho):
                return z

#Standard normal distrubution (mu=0, sigmal=1) truncated on interval [a,b]
def tnorm_std(a, b):
    #print a,b
    if(a>b):
        raise Exception("tnorm: Truncation interval is empty!")
    elif(a==b):
        return a
    elif(a*b<0):    #Case 1
        return tuvnorm1(a,b)
    elif(a>=0):     #Case 2
        return tuvnorm2(a,b)
    else:           #Case 3
        return -1*tuvnorm2(-b,-a)

#General normal distrubution truncated on interval [a,b]
def tnorm(mu,sigma, a, b):
    a2 = (a-mu)/sigma
    b2 = (b-mu)/sigma
    r2 = tnorm_std(a2,b2)
    r = mu+sigma*r2
    #In exact arithmatic, the following conditions are always statisfied.
    #However, for very large values, roundoff errors may leads to violations
        .
    if(r<a):
        r=a
    if(r>b):
        r=b
    return r
```

APPENDIX C

Blocking Method: Estimating Mean's Error

The SMS method approximates the mean of a distribution using a finite population[1] and thus the computed mean differs from the actual one by an amount that diminishes as the population size increases. We can estimate this error using the standard deviation of the population mean; an easy task when the samples are uncorrelated. However, since our samples are drawn using Gibbs sampling (a Markov chain), they are correlated and using the uncorrelated formula would give very small and misleading error estimates. To get reliable error estimates, we use the *blocking method* described in Ref. [26].

Let us suppose, we have n samples drawn from a probability distribution with mean μ and variance σ^2, then we can approximate the mean μ using the average m

$$\mu \approx m \equiv \frac{1}{n}\sum_{i=1}^{n} x_i \ , \tag{C.1}$$

and the variance of m reads

$$\sigma^2(m) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{Cov}(x_i, x_j) \ . \tag{C.2}$$

where $\mathrm{Cov}(x_i, x_j)$ is the covariance between sample i and sample j.

[1] Population means a set of samples drawn from some probability distribution. They can be correlated or uncorrelated.

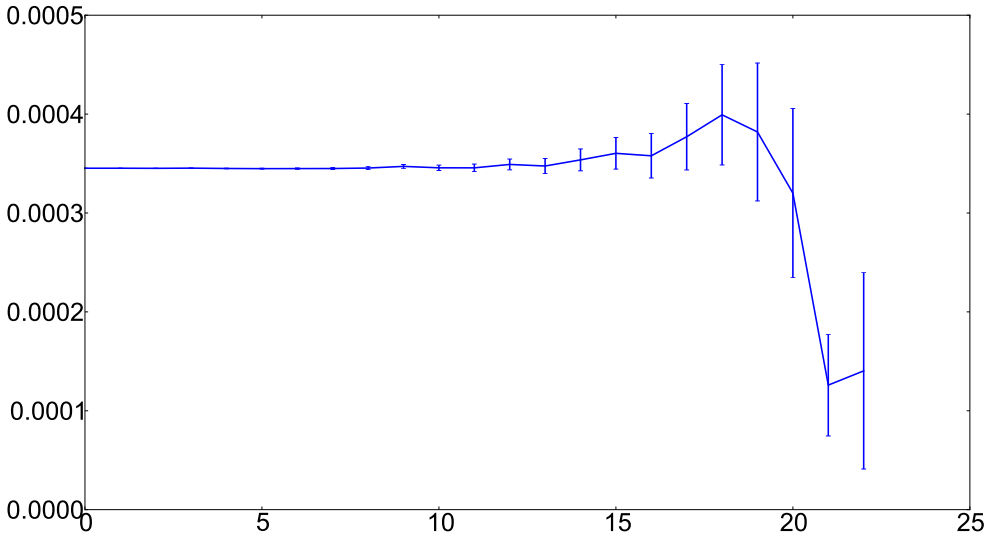

Fig. C.1.: Estimate of $\sigma(m)$ against the number of block transformations for samples drawn from the standard normal distribution using typical random library procedure. Notice that we are at the plateau from the very beginning because the samples are uncorrelated.

Uncorrelated Case When the samples x_i are uncorrelated, the covaraince matrix is diagonal. Since the samples are drawn from the same probability distribution, the diagonal elements are identical and equal the variance of the probability distribution. Therefore, the previous expression reduces to:

$$\sigma^2(m) = \frac{1}{n^2}\sum_{i=1}^{n} \mathrm{Cov}(x_i, x_i) = \frac{1}{n}\sigma^2 \,. \tag{C.3}$$

To estimate σ^2, we use the unbiased variance estimator

$$\sigma^2 \approx \frac{1}{n-1}\sum_{i=1}^{n}(x_i - m)^2 \,. \tag{C.4}$$

Substituting Eq. (C.4) in Eq. (C.3), we get

$$\sigma^2(m) \approx \frac{1}{n(n-1)}\sum_{i=1}^{n}(x_i - m)^2 = \frac{1}{n(n-1)}\sum_{i=1}^{n}(x_i^2 + m^2 - 2mx_i) \,. \tag{C.5}$$

But $n \cdot \sum_{i=1}^{n} x_i = m$ according to Eq. (C.1), so

$$\sigma^2(m) \approx \frac{1}{n(n-1)}\sum_{i=1}^{n} x_i^2 - \frac{1}{n-1}m^2 \,, \tag{C.6}$$

and the standard deviation of the mean of a population of uncorrelated samples reads

$$\sigma(m) \approx \frac{1}{n-1}\left[\frac{\sum_{i=1}^{n} x_i^2}{n} - \frac{(\sum_{i=1}^{n} x_i)^2}{n^2}\right] \,. \tag{C.7}$$

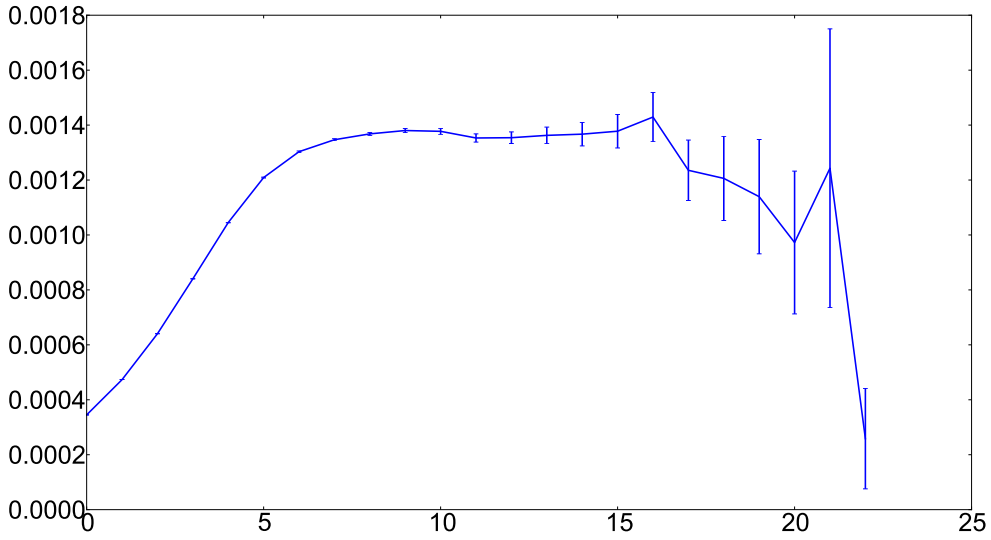

Fig. C.2.: Estimate of $\sigma(m)$ against the number of block transformations for samples drawn from the standard normal distribution using Metropolis algorithm with step size $\Delta = 1$. Notice that we reach the plateau after about 10 block transformations.

Correlated Case For the general case of correlated samples, a direct approach to get $\sigma(m)$ is to estimate the covariance matrix between different samples and utilize Eq. (C.2). The blocking method avoids the calculation of the covariance and it goes as following. We divide the samples into "blocks" of size 2 and compute the average of each block. This gives a new set of samples $x'_i = x_{2i} + x_{2i+1}$ where $i = 1, .., N/2$. We repeat this transformation recursively and at each time we compute $\sigma(m)$ using x'_i as if they were independent using Eq. (C.7). As the block size increases (it doubles each time), $\sigma(m)$ increases but after enough number of transformations (when the block size is larger than correlation time), it reaches a fixed point and does not change by further blocking. This fixed point of $\sigma(m)$ is the "true" value we are looking for. Further blocking will eventually lead to fluctuating values of $\sigma(m)$ because the number of blocks gets smaller and the estimator becomes inaccurate.

The proof that repeated blocking gives less correlated samples can be found in Ref. [26]. Assuming that this is true, we can give a simple argument why a plateau should exists. If we apply the blocking to a population of uncorrelated samples, we will obtain a new set of samples of half the size. Each new sample has half the original variance because

$$\mathrm{Var}\left(\frac{x_i + x_j}{2}\right) = \frac{1}{4}\mathrm{Var}(x_i + x_j) = \frac{1}{4}\mathrm{Var}\,(x_i) + \frac{1}{4}\mathrm{Var}\,(x_j) = \frac{\sigma^2}{2} \tag{C.8}$$

Besides, the new set of samples has the same average m. Now we compute the variance of the mean of the new set using Eq. C.3

$$\sigma'^2(m) = \frac{1}{n/2}\sigma^2/2 = \sigma^2(m)\ , \tag{C.9}$$

which gives the same value we obtain from the original set of samples. This means that once the samples we obtain by repeated blocking are uncorrelated, the estimate of the $\sigma(m)$ (assuming we have enough number of samples) reaches a fixed point and stays the same upon further blocking.

Applying the method is easy. Plot the standard deviation $\sigma_p(m)$ against the number of transformations $p = \log_2(b)$, where b is the block size, and look for a plateau in the plot. Ref. [26] provides the following error estimate for $\sigma(m)$

$$\text{Error in } \sigma(m) \approx \pm \frac{\sigma(m)}{\sqrt{2(n-1)}} \tag{C.10}$$

which is used to compute error bars in the previous plot. We will also use it later in detecting the plateau automatically.

As an illustrating example, we generate three sets of samples drawn from a standard normal distribution. Each set has 2^{23} samples and the sets differ in the amount of correlation. In the first set, the samples are generated using a common library procedure for generating normal random variables. In the second and third sets, we generate the samples using Metropolis algorithm where the suggested moves lie uniformly in the interval $[-\Delta, +\Delta]$ with $\Delta = 1$ and $\Delta = 0.1$, respectively. For the first case (see Fig. C.1), the samples are completely uncorrelated and thus we are at the plateau from the very beginning. In the second and third cases, the Metropolis algorithm is a Markov chain and that generates correlated samples. The correlation is smaller for the second case than for the third, because in the second one we allow moves up to one standard deviation, while the allowed moves are smaller for the third case and thus the sampling is less efficient. This difference in correlation time is reflected in (see Fig. C.2) and (see Fig. C.3) where we see that the plateau is reached after about 9 transformations for the second set while it needs about 14 transformations for the third set.

Detecting the plateau In the examples, we had to plot $\sigma(m)$ against the number of block transformations and detect the plateau by looking at the plot checking whether it is flat within errorbars. However, it is desirable to have an automatic procedure of detecting the plateau.

We propose the following criterion. We are at the plateau when $\sigma(m)$ as a function of the number of block transformations is constant. A constant function is a function whose all derivatives are zero. Practically, we found that it is sufficient to check only the first two derivatives. So the detecting procedure is as following:

Scan points from left to right. For each point, evaluate the first and second derivatives using central-difference formulas. If their value is less than the error estimate at that point, Eq. (C.10), then we are at the plateau and the value of $\sigma(m)$ at that point is our best estimate. The last few points (say 3 or 4 points) should excluded be from the scan because their values and the error estimate of their values are unreliable as they are calculated with very few number of samples. If the criterion is not satisfied for any point, then the plateau does not exist and more samples are needed.

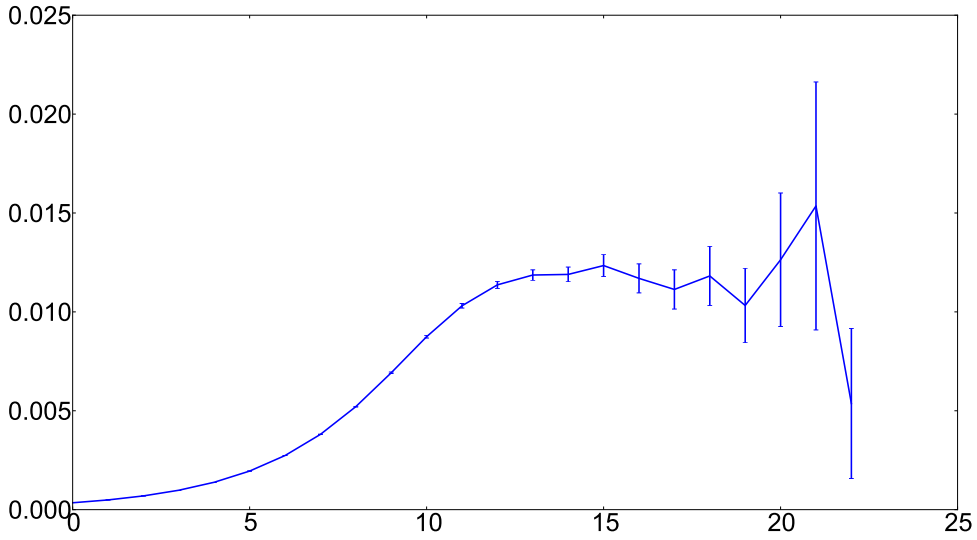

Fig. C.3.: Estimate of $\sigma(m)$ against the number of block transformations for samples drawn from the standard normal distribution using Metropolis algorithm with step size $\Delta = 0.1$. Notice that we reach the plateau after about 15 block transformations.

Implementation

Ref. [27] describes an efficient algorithm and data structure for implementing the blocking method. A python code of this algorithm combined with our method of detecting the plateau is provided below.

```
from __future__ import division
from matplotlib.pylab import *
from numpy import *

#Helping Class
class Statistic(object):
    def __init__(self):
        self.num = 0
        self.sum = 0.0
        self.sumSq = 0.0
        self.waitingSample = None
    def addSample(self, sample):
        self.num += 1
        self.sum += sample
        self.sumSq += sample**2
        if(self.waitingSample == None):
            self.waitingSample = sample
            return None
        else:
            avgSample = (self.waitingSample+sample)/2.0
            self.waitingSample = None
```

```
            return avgSample

    def getMean(self):
        return self.sum/self.num

    #mean's standard deviation
    def getS(self):
        return sqrt((self.sumSq/self.num - (self.sum/self.num)**2)/(self.num
            -1))

    #error in mean's standard deviation
    def getSErr(self):
        return self.getS()/sqrt(2*(self.num-1))

#Main Class.
#Simply add the samples as they arraive using addSample().
#At any time, call getMean() and getS() to get the samples mean and its
#standard deviation.
#Each sample is assumed to be a numpy array.
class Decorrelation(object):

    def __init__(self):
        self.samplesN = 0
        self.stats = list()
        self.lastSample = None
        #how many several consecutive points should statisfy the zero
        #derivatives condtion
        self.plateauLenght = 1

        #how many values to discard from the end
        self.discard = 4

    #we assume that the samples are numpy arrays
    def addSample(self, sample):
        self.lastSample = sample
        self.samplesN += 1
        for stat in self.stats:
            sample = stat.addSample(sample)
            if(sample==None):
                break
        if(sample!=None):
            newStat = Statistic()
            newStat.addSample(sample)
            self.stats.append(newStat)

    #return the mean of the samples
    def getMean(self):
        #all stats should theortically give the same mean
        #(when the number of samples in a power of two). However, the last
            one
        #is however more numerically stable since summation is done pairwise
            .
        return self.stats[-1].getMean()

    #return best estimate of mean's standard deviation for all components
    #when the plateau is not found, a nan value is returned.
```

```
    def getS(self):

        sampleSize = len(self.lastSample)
        isPlateau = zeros(shape=(sampleSize,len(self.stats)), dtype=bool)

        for j in range(1,len(self.stats)-self.discard-1):
            Sj = self.stats[j].getS()
            Sp = self.stats[j+1].getS()
            Sm = self.stats[j-1].getS()
            derv1= abs(Sp-Sm)/2.0
            derv2= abs(Sp-2*Sj+Sm)
            eps = self.stats[j].getSErr()
            isPlateau[:,j] = logical_and(less(derv1, eps), less(derv2, eps))
            shift = 1
            while(shift<self.plateauLenght & (j-shift)>=0):
                isPlateau[:,j-shift] = logical_and(isPlateau[:,j-shift],
                    isPlateau[:,j])
                shift+=1

        s = ones(sampleSize)*nan
        for j in range(1,len(self.stats)-self.discard-self.plateauLenght):
            Sj = self.stats[j].getS()
            for i in range(sampleSize):
                if(isPlateau[i,j] and isnan(s[i])):
                    s[i] = Sj[i]

        return s

    #plot mean's standard deviation with error bars for a specific
    #samples' compnent
    def plotS(self, i):
        mdevs = list()
        mdevErrs = list()
        #last stats contains only one sample so no variance
        for j in range(len(self.stats)-1):
            mdevs.append(self.stats[j].getS()[i])
            mdevErrs.append(self.stats[j].getSErr()[i])

        errorbar(range(len(self.stats)-1) , mdevs, yerr=mdevErrs)
        show()
```

Bibliography

[1] R. Kress, *Linear Integral Equations* (Springer, Berlin, 1989).

[2] J. Waldvogel, Approximation and Computation, edited by W. Gautschi, G. Mastroianni, T.M. Rassias (Springer, New York, 2011), pp 267-282, Towards a General Error Theory of the Trapezoidal Rule.

[3] C. Schwarz, Numerical Integration of Analytic Functions,
J. Comput. Phys. **4**, 19-29 (1969).

[4] G.H. Golub, C.F. Van Loan, *Matrix Computations, Third Edition*
(The Johns Hopkins University Press, Baltimore and London, 1996).

[5] A.N. Tikhonov, V.Y. Arsenin, *Solution of Ill-posed Problems* (Winston & Sons, Washington, 1977).

[6] C.W. Groetsch, *The theory of Tikhonov regularization for Fredholm equations of the first kind*, Research Notes in Mathematics 105 (Pitman, Boston, 1984).

[7] P.C. Hansen, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Review **34**, 561-580 (1992).

[8] C.L. Lawson, R.J. Hanson, *Solving Least Squares Problems*
(SIAM, Englewood Cliffs, New Jersey, 1974).

[9] A.W. Sandvik, Phys. Rev. B **57**, 10287 (1998).

[10] K. Vafayi, O. Gunnarsson, Phys. Rev. B **76**, 035115 (2007).

[11] O.F. Syljuasen, Phys. Rev. B **78**, 174429 (2008).

[12] S. Geman, D. Geman, *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **6** 721-741 (1984).

[13] G. Casella, E.I. George, *Explaining the Gibbs sampler*,
The American Statistician **46**, 167-174 (1992)

[14] **Noisy kernel** method is a recipe we found before developing the SMS method. It is a simple way of sampling the space of models. Obtaining one sample is done by adding noise to the entries of the matrix and then solving the non-negative least squares problem using this noisy kernel. The final solution is the average over a larger set of samples. The noise added to kernel element $\mathbf{K}_{i,j}$ is normally distributed with zero mean and a standard deviation of $\sigma\mathbf{K}_{i,j}$ where σ is usually ten times larger than the standard deviation of te data noise. Tests showed that this method gives similar results to non-negative Tikhonov. The results, however, are slightly better because the solution values approach zero smoothly unlike non-negative Tikhonov solutions which tend to be clamped.

[15] W.C. Horrace, *Some results on the multivariate truncated normal distribution*,
Journal of Multivariate Analysis **94**, 209-221 (2005).

[16] O. Gunnarsson, M.W. Haverkort, and G. Sangiovanni, Phys. Rev. B **82**, 165125 (2010).

[17] O. Gunnarsson, M.W. Haverkort, and G. Sangiovanni, Phys. Rev. B **81**, 155107 (2010).

[18] J.W. Negele, H.Orland, *Quantum many-particle systems*
(Westview Press, Colorado, 1998).

[19] P. Nozieres, *The Theory of Interacting Fermi Systems*
(Westview Press, Colorado, 1997).

[20] E.N. Economou, *Green's Functions in Quantum Physics*
(Springer, Berlin, 1983), 2nd ed.

[21] A.M. Tremblay, *A refresher in many-body theory*, (Online document, updated May 2008, cited April 2013), Available from URL: `www.physique.usherbrooke.ca/tremblay/cours/phy-892/jouvence.pdf`

[22] J.O. Fjaerestad, *Introduction to Green functions and many-body perturbation theory* (Online document, updated March 2013, cited April 2013), Available from URL: `www.nt.ntnu.no/users/johnof/green-2013.pdf`

[23] C.P. Robert, *Simulation of truncated normal variables*,
J. Statistics and Computing **5**, 121-125 (1995).

[24] L. Devroye. *Non-Uniform Random Variate Generation*
(Springer-Verlag, New York, 1985).

[25] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing* (Cambridge University Press, New York, 2007)

[26] H. Flyvbjerg, H.G. Petersen, J. Chem. Phys. **91**, 461 (1989).

[27] D.R. Kent IV *et al.*, J. Comput. Chem. **28**, 14 (2007).